Universidad Nacional de Educación a Distancia (UNED)
Escuela Técnica Superior de Ingeniería Informática

Diagnóstico Automático de la Figura Compleja de Rey mediante Redes Siamesas

Autor: Eladio Estella Nonay
Director: Mariano Rincón Zamorano

Máster Universitario en Inteligencia Artificial Avanzada:
Fundamentos, Métodos y Aplicaciones

Trabajo Fin de Máster
Septiembre 2020

Índice

Diagnóstico Automático de la Figura Compleja de Rey mediante Redes Siamesas

Eladio Estella Nonay, Mariano Rincón Zamorano

Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia - UNED, Madrid, España

Resumen

El uso de las últimas técnicas en Inteligencia Artificial como herramienta de ayuda en el ámbito médico es un reto actual e irrenunciable. En este contexto, los métodos de detección de similitudes y las Redes Neuronales Convolucionales utilizadas en tareas de visión artificial para la extracción de características pueden contribuir enormemente al análisis de pruebas médicas basadas en dibujos a mano alzada. Este trabajo reúne ambas ideas y presenta la utilización de las Redes Neuronales Siamesas para realizar un diagnóstico automático de enfermedades neurodegenerativas basado en la prueba de la Figura Compleja de Rey. Profundiza en la idoneidad de este tipo de redes y realiza un estudio comparativo de 3 arquitecturas diferentes: una Red Neuronal Artificial, una Red Neuronal Siamesa y una modificación de ésta para su entrenamiento mediante *triplets*. Para ello, se dispone de cerca de 500 dibujos recogidos en un estudio de investigación en el campo de la neuropsicología, realizados por pacientes sanos o con algún grado de deterioro cognitivo. Debido al reducido número de instancias, se propone el preentrenamiento de las redes con la técnica de *Transfer Learning* mediante un *dataset* de dibujos mucho mayor y de características similares.

Palabras clave: Prueba Figura Compleja de Rey, Redes Neuronales Siamesas, Triplets, Deterioro Cognitivo Leve

1. Introducción

El aumento de la esperanza de vida en las últimas décadas ha supuesto un incremento en los casos de enfermedades neurodegenerativas asociadas principalmente a la edad. Enfermedades que hasta mediados del siglo pasado se consideraban simplemente parte del proceso de envejecimiento. El hecho de que la esperanza de vida siga creciendo cada año, ha impulsado a la comunidad científica a dirigir gran cantidad de recursos a la investigación de enfermedades neurodegenerativas relacionadas con el deterioro cognitivo. Sin embargo, aunque se lleva trabajando desde los años 70 en este tipo de enfermedades, no se dispone de tratamientos efectivos. Es ahí donde el diagnóstico precoz de este tipo de trastornos juega un importante papel a la hora de poder minimizar sus efectos. Se trata de un reto tanto médico, para encontrar nuevos métodos seguros y fiables, como tecnológico, para encontrar

maneras de colaborar con la ciencia médica aportando los últimos avances como herramientas. Sólo de esta forma, estas enfermedades serán sostenibles en un futuro a nivel económico, sanitario y social.

Una de las pruebas capaces de ayudar en el diagnóstico de enfermedades neurodegenerativas es la prueba de la Figura Compleja de Rey (FCR) [1], en la cual un paciente debe realizar dos tareas. En la primera, realiza una copia a mano alzada de dicha figura. En la segunda, se retira la FCR y el sujeto reproduce la figura de memoria. Actualmente los expertos pueden realizar dos tipos de análisis de las copias realizadas. En el primero se realiza un análisis cuantitativo [2], en donde la figura se divide en 18 elementos y a cada uno se le asigna una puntuación en función de su presencia, deformación y localización (Figura 1). La asignación de los puntos se realiza según los criterios expuestos en la Tabla 1 para cada uno de los 18 elementos, y su suma sirve como indicador para el diagnóstico del paciente. La media para una persona adulta es de 32 puntos [3]. En el segundo, se realiza un análisis cualitativo de la copia [4], que atiende a criterios como la posición del dibujo en la hoja, proporciones de elementos, rotaciones u omisiones. Ambos análisis suponen el consumo de gran cantidad de recursos personales, temporales y sanitarios, y añade además la subjetividad del experto durante la evaluación para criterios como la deformación, la proporción o la correcta localización de un elemento.

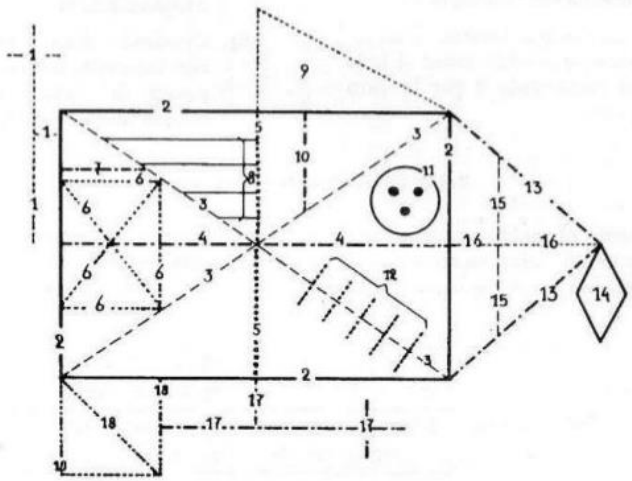


Figura 1. FCR dividida en los 18 elementos puntuables.

Grado de similitud	**Puntos**
Parte correcta bien situada	2
Parte correcta y mal situada	1
Parte deformada (pero reconocible y bien situada)	1
Parte deformada (pero reconocible y mal situada)	0'5
Parte irreconocible o ausente	0

Tabla 1. Asignación de puntos para cada elemento dependiendo de su grado de similitud con el modelo.

El objetivo de este trabajo se centra en generar un diagnóstico médico automático a partir de la copia realizada en la primera tarea de la prueba de FCR. Se busca proporcionar al personal médico nuevas

herramientas con las que poder obtener resultados fiables que optimicen los recursos del sector y mejoren el poder de diagnóstico. Se tiene en cuenta además que no existen restricciones temporales, al no ser una aplicación en tiempo real, ni de memoria, al no tratarse de una aplicación móvil.

Actualmente las técnicas más prometedoras para la creación de sistemas automáticos se enmarcan en el ámbito del *Deep Learning* (DL). Uno de sus métodos más exitosos son las Redes Neuronales Convolucionales (en inglés CNNs - Convolutional Neural Networks) [5], las cuales han sido clave para el desarrollo de aplicaciones dentro del campo de la visión artificial. Estas redes están divididas por niveles, en los cuales se calculan características de la imagen cuyo nivel de abstracción aumenta con la profundidad del nivel. El resultado de estas redes se puede presentar en forma de vector e interpretarlo como una representación compacta de la imagen, también llamada *embedding*. Las Redes Siamesas (RSs) utilizan esta idea para, a partir de dos imágenes, extraer sus características, compararlas e identificar si se trata del mismo objeto. Por ello, se propone la utilización de RSs para la generación automática de un diagnóstico en base a las diferencias encontradas entre el dibujo realizado por el paciente y la FCR. Este trabajo plantea la comparación de 3 arquitecturas que utilizan como pieza central un extractor de características basado en CNNs. La primera arquitectura añade al extractor una Red Neuronal Artificial (RNA) para la tarea de clasificación, la segunda utiliza una arquitectura de RS y la tercera una modificación de la segunda utilizando el entrenamiento mediante *triplets*.

Respecto a los datos, se dispone de un *dataset* que contiene dibujos procedentes de pruebas reales y no superan las 500 instancias. Para abordar el problema de la limitación del tamaño del *dataset*, se propone la utilización de *Transfer Learning* (TL) preentrenando el extractor con una base de datos (BD) mucho mayor, similar a la primera en cuanto a la forma y trazos de los dibujos.

La contribución de este trabajo se resume de la siguiente manera: la aplicación en el ámbito médico de los últimos métodos desarrollados en RSs para la detección de posibles signos que indiquen un deterioro cognitivo y la generación del correspondiente diagnóstico de forma automática. La hipótesis de partida de esta investigación sugiere que las RSs, dadas sus características, son adecuadas para esta tarea.

2. Trabajo relacionado

2.1 Aplicaciones de visión artificial sobre dibujos a mano alzada

Los trabajos que analizan dibujos de forma automática mediante técnicas de DL se engloban en dos categorías: los que se centran en tareas de clasificación [6] y las aplicaciones de recuperación de imágenes basadas en dibujos (en inglés SBIR - *sketch-based image retrieval*) [7] [8]. Los primeros asignan a una imagen de entrada una etiqueta que indica la pertenencia a una clase. Los segundos se componen de una BD de imágenes reales y a partir del análisis de un dibujo de entrada, devuelven la imagen real más parecida encontrada en la BD. Esta combinación de dibujos e imágenes reales (*cross domain applications*) se puede encontrar ya a día de hoy en aplicaciones comerciales, en las que el usuario realiza un boceto de la apariencia de un producto y recibe la información del objeto a la venta más parecido.

Para el desarrollo de dichas aplicaciones, se encuentran disponibles diferentes tipos de *datasets* que contienen dibujos a mano alzada. Algunos de ellos presentan parejas de bocetos con imágenes reales correspondientes a la misma clase (*cross-domain methods* [9]) y que están enfocados a las aplicaciones SBIR para tareas de identificación de caras [7], zapatos y sillas [8], o de clases más generales como en el caso de "Sketchy" (Figura 2) [6]. Existen además *datasets* que contienen solamente representaciones

a mano alzada para el análisis, identificación o clasificación de elementos, como pueden ser caracteres ("*Omniglot*"[10]), dígitos ("MNIST" [11]) y objetos y seres vivos en general ("*Sketch*" [12], "*QuickDraw*" [13]) (Figura 3). Todas sus instancias tienen varios rasgos en común:

1- contienen la representación manuscrita de un concepto, idea o símbolo;

2- han sido realizadas por un sujeto con ayuda de un soporte electrónico;

3- han sido procesadas para eliminar ruido u otros elementos no deseados;

4- el resultado final se expresa en un mapa de bits en escala de grises.

Una característica de la mayoría de estos *datasets* es su pequeño tamaño (Tabla 2) en comparación a los *datasets* de imágenes reales más utilizados como Open Images Dataset (~9.000.000), ImageNet (~1.500.000 instancias) o MS-COCO (330.000).

En ambas categorías de análisis de dibujos se puede recurrir a técnicas clásicas de clasificación, en donde la salida del sistema es una distribución de probabilidad que indica la posible pertenencia de la entrada a cada una de las clases previamente entrenadas. Estos métodos necesitan miles de instancias por clase para crear un espacio de salida en función de las características observadas, y la distribución de probabilidad es difícil de ajustar mediante el entrenamiento cuando existen clases con características muy parecidas (p.e. perro/lobo).

Para solventar estas carencias, los esfuerzos en los últimos años se han centrado en la aplicación de técnicas de detección de similitudes [14]. Estos métodos son capaces de calcular diferencias entre distintas instancias y con ello determinar la pertenencia a la clase correspondiente. Dentro de estas técnicas se presentan las aplicaciones de *one-shot learning* [15], capaces de aprender la información de cada categoría simplemente a partir de unas pocas instancias por clase y determinar el grado de similitud entre ellas. Mediante el cálculo de características para una instancia, se puede alcanzar una representación más compacta y fácilmente comparable con otras instancias similares. En este sentido, toma especial relevancia una adecuada generalización de las características mediante un apropiado proceso de extracción, que dependerá de los tipos de datos y la tarea a realizar. En el caso de la visión artificial, los extractores de características típicos y más exitosos se encuentran en el campo de las CNNs [16] [17].

Desde el punto de vista teórico, el paradigma de detección de similitudes mediante la estrategia *one-shot learning* encaja perfectamente como solución al diagnóstico en la prueba de la FCR:

- La detección de similitudes entre un dibujo y la FCR responde fielmente como criterio discriminatorio de pertenencia a la clase de personas sanas (representada siempre por la FCR).
- Esta comparación puede realizarse con *embeddings* que representen de forma compacta las características de cada dibujo.
- El entrenamiento de sistemas de clasificación clásicos basados en RNAs requieren miles de instancias. El pequeño tamaño del *dataset* disponible no resulta un inconveniente para los sistemas basados en *one-shot learning*.
- Los extractores de características que forman parte de la arquitectura de las RSs pueden entrenarse de manera independiente con *datasets* masivos de instancias semejantes.

Figura 2. Ejemplos de parejas de dibujo-imagen real para aplicaciones SBIR. "Sketch Me That Shoe" (izda), "Sketchy" (dcha).

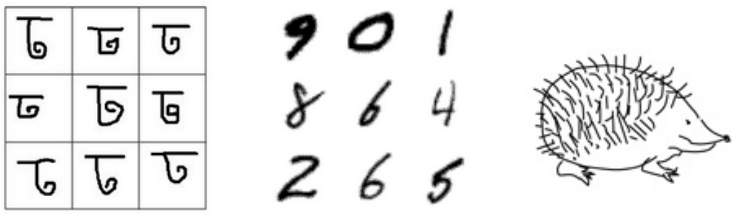


Figura 3. De izquierda a derecha, ejemplos de imágenes en los *datasets* "*Omniglot*" [10],"MNIST" [11] y "*Sketch*" [12].

Dataset	Clases	Instancias
Omniglot	1.623	32.460
MNIST	10	70.000
Sketch	250	20.000
Sketch Me That Shoe	2	419+217
Sketchy	125	75.471
QuickDraw	345	~50.000.000

Tabla 2. Relación de clases e instancias para cada *dataset* orientado a dibujos.

2.2 Redes Siamesas

Las RNAs se han implantado con éxito en diferentes campos como el comercio electrónico [18], automoción [19] o meteorología [20], así como dentro del ámbito médico mediante estudios teóricos [21] o prácticos en temas como la asistencia en resonancias magnéticas [22], mamografías [23], detección de enfermedades mediante análisis de sangre [24] o deterioro cognitivo [25] [26] [27]. Para todos ellos, el desarrollo de las RNAs durante esta última década ha supuesto un pilar fundamental y ha abierto un amplio abanico de posibilidades para el análisis de datos que hasta hace unos años estaba muy limitado tecnológicamente. En este campo se enmarcan las RSs, que se presentan como el mayor exponente para aplicaciones *one-shot learning*.

La motivación en el desarrollo de las RSs se fundamenta en la capacidad del ser humano para aprender representaciones abstractas que aplica posteriormente sobre elementos nunca antes percibidos. Esta capacidad permite la asociación entre representaciones y, consecuentemente, hace que nuevos elementos sean reconocidos o clasificados [28] [29]. Esta forma de procesado de información se ajusta especialmente a tareas de visión artificial.

Las RSs se presentaron por primera vez en 1993 para la validación de firmas de forma automática [30]. El sistema se componía de dos ramas, que recibían como entrada una firma original y otra firma presumiblemente del mismo autor, las analizaba y confirmaba su autenticidad. Se necesitaba por lo tanto una sola firma original de cada usuario para la detección de falsificaciones. Posteriormente, las RSs se han abierto camino en campos que trabajan con imágenes reales, como el seguimiento de objetos [31] o el reconocimiento facial [32]. En el ámbito médico, las aplicaciones de RSs utilizan imágenes y se agrupan principalmente en 3 tipos de tareas. Las primeras se centran en el diagnóstico de la enfermedad, como por ejemplo el Alzeihmer [33], las segundas evalúan el nivel de gravedad de la enfermedad, como en el caso de enfermedades relacionadas con la artritis [34], y las últimas son aplicaciones de recuperación de imágenes basadas en el contenido (en inglés CBMIR - *Content-based medical image retrieval*) como las utilizadas en el campo de la retinopatía diabética [35].

La aplicación de las RSs sobre dibujos o representaciones manuscritas también ha sido objeto de estudio tanto para el reconocimiento de caracteres [36] como para el reconocimiento de conceptos a partir de dibujos. La utilidad de esta última se centra en aplicaciones SBIR, que devuelven imágenes detalladas de objetos en 2D [37] o 3D [38]. Algunos trabajos proponen además sistemas entrenados mediante *triplets* [39], los cuales mejoran los resultados para las RSs.

Si bien para la extracción de características en dibujos existen técnicas relacionadas con el análisis del gradiente [40] o la cadena de trazos [41], las CNNs han sobrepasado notablemente a las técnicas anteriores y están firmemente implantadas en tareas SBIR para aplicaciones comerciales a través de CNNs profundas [9] [42].

Por tanto, aunque no se conocen publicaciones sobre el empleo de RSs para aplicaciones de detección de enfermedades cognitivas a través de dibujos, este trabajo apuesta por la implantación de las RSs para su diagnóstico mediante la extracción de características de dibujos con CNNs.

3. Materiales y métodos

3.1 Datasets utilizados

El *dataset* inicial [43] está formado por dibujos de la prueba de la FCR recogidos en diferentes sesiones de evaluación neuropsicológica durante el estudio de investigación presentado en [44]. Este estudio ha sido realizado en centros culturales por sujetos inicialmente sanos y comprende la evaluación y el seguimiento anual de sus capacidades cognitivas mediante una batería de pruebas. El número de evaluaciones por sujeto varía entre 1 y 5. En función de los resultados obtenidos en las pruebas, a un sujeto se le asigna el perfil de Diagnóstico Cognitivo Leve (DCL) cuando en dos o más pruebas obtiene una puntuación que está al menos 1'5 desviaciones típicas por debajo de la media de su grupo correspondiente (según edad y formación educativa). En caso contrario el sujeto es clasificado como sano. El estudio distingue además 3 tipos de DCL:

1- DCL amnésico (DCLa): la puntuación es inferior al umbral anteriormente presentado en al menos

dos pruebas del Test de Aprendizaje Verbal España-Complutense (TAVEC) [45];

2- DCL no amnésico (DCLna): la puntuación es inferior al umbral anteriormente presentado en al menos dos pruebas no recogidas en el TAVEC;

3- DCL multidominio (DCLm): la puntuación es inferior al umbral anteriormente presentado en una prueba del TAVEC y en al menos otra no recogida en él.

Los dibujos del *dataset* inicial han sido escaneados (Figura 4) y preprocesados mediante la eliminación manual de anotaciones y números que aparecían al lado de los dibujos como notas informativas. El resultado es un conjunto que cuenta con 887 instancias de las que solamente 477 están etiquetadas. Las etiquetas se han asignado siguiendo la clasificación de los sujetos en el estudio anterior (sano, DCLa, DCLna, DCLm).

El *dataset* utilizado en este trabajo (*dataset* de la Figura Compleja de Rey, DFCR) es un subconjunto del *dataset* inicial que recoge todas las instancias etiquetadas, las cuales han sido divididas en dos clases: los dibujos realizados por personas sanas (clase "sano") y los realizados por personas con diferentes grados de deterioro cognitivo (DCLa, DCLna, DCLm; clase "no sano"). Las instancias de ambas clases se distribuyen en un 58'5% para "sanos" y un 41'5% para "no sanos".

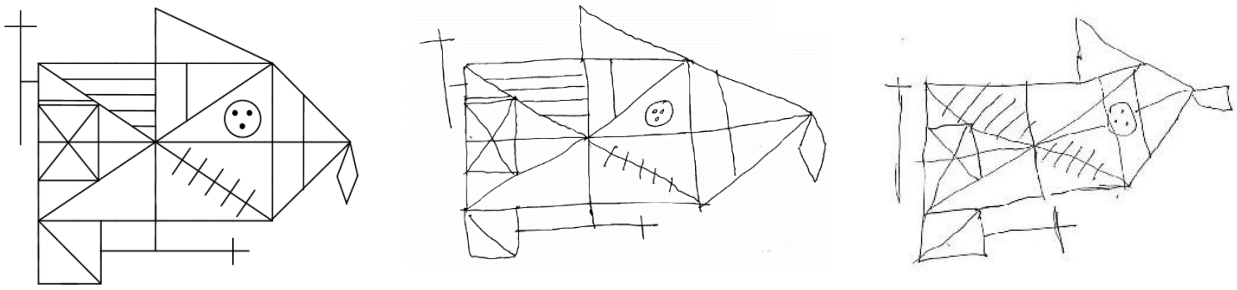


Figura 4. FCR (izda), ejemplo de dibujo realizado por un sujeto sano (centro) y por un sujeto con deterioro cognitivo (dcha).

Debido al reducido tamaño del *dataset* a analizar, se utiliza la estrategia de TL para el preentrenamiento de las distintas arquitecturas. El *dataset* utilizado en esta fase es *QuickDraw* (Figura 5) debido a su enorme tamaño y a que las diferentes clases generalizan de forma adecuada las representaciones de dibujos, donde se requieren diferentes tipos de trazos para rectas y curvas con diferentes longitudes y ángulos. Además, *QuickDraw* permite la descarga de la representación temporal de la secuencia de los trazos. Aunque no es necesaria en este trabajo al no contar con esta información en el *dataset* a analizar, podría ser de utilidad en futuras aplicaciones.

QuickDraw tiene su origen en 2016 como juego online. En él, un usuario dibujaba un objeto y otro debía adivinar de qué se trataba. La idea original se modificó manteniendo la premisa de que un usuario dibujase un objeto, pero esta vez una red neuronal intentaba reconocer a qué clase pertenecía, es decir, se añadió la funcionalidad de reconocimiento automático. En 2017 el equipo Magenta de Google Research aprovechó la aplicación para poder crear una base de datos utilizando las aportaciones de los usuarios del juego. También se les insta a participar de forma activa indicando errores tales como un etiquetado incorrecto o dibujos que no corresponden al concepto a representar. Existen diferentes formatos descargables para las imágenes, como raw data, 28x28 mapa de bits o las imágenes originales

con extensión "png", siendo este último el formato elegido para el entrenamiento. Se trata por lo tanto de una base de datos de dibujos accesible y descargable de forma gratuita a través de la web. Se compone de 345 clases con alrededor de 50 millones de instancias generadas por más de 15 millones de jugadores.

Figura 5. Ejemplos de dibujos del *dataset "QuickDraw"*.

Los dibujos han sido descargados como imágenes en escala de grises con valores entre 0 y 255 y 255x255 píxeles de tamaño. Para restringir el tamaño y tiempo de entrenamiento se ha elegido un *subset* del *dataset* global (SQD, *subset* de *QuickDraw*) que comprende únicamente 1000 instancias de 19 clases diferentes (total 19.000 instancias). La elección de las clases se ha basado en la generalización de diferentes formas, figuras y trazos. La Figura 6 muestra ejemplos de clases cuyos dibujos contienen normalmente líneas rectas (clase "espada"), líneas curvas ("hamburguesa"), una mezcla de ambas ("avión"), líneas en zig-zag ("sierra"), líneas paralelas e intersecciones ("banco" y "violín"), círculos ("calavera"), cuadrados ("calculadora"), triángulos ("estrella") o que representan formas complejas ("pingüino").

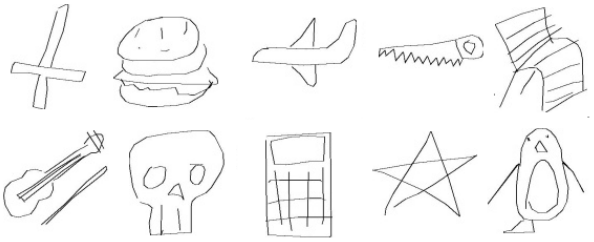


Figura 6. Ejemplos de instancias del *dataset* utilizado en la fase de TL. Las clases representadas son, en la fila de arriba, de izda a dcha: "espada", "hamburguesa", "avión", "sierra", "banco". En la fila de abajo: "violín", "calavera", "calculadora", "estrella", "pingüino".

Todas las imágenes utilizadas en este trabajo han sido reescaladas previamente a un tamaño de 100x100 píxeles, lo que reduce considerablemente el tamaño de entrada al sistema manteniendo el detalle de los dibujos. Las intensidades en escala de grises de todos los píxeles en cada imagen se han normalizado entre 0 y 1.

3.2 Aplicación de las Redes Siamesas al análisis de la prueba de la FCR

Como se ha comentado en secciones anteriores, las RSs son una variante de las redes neuronales capaces de encontrar similitudes y relacionar conjuntos de datos de entrada. Su arquitectura consiste en dos redes gemelas o ramas, que comparten el valor de los pesos, los cuales se actualizan simultáneamente durante el proceso de entrenamiento. Cada una de las ramas equivale a un extractor de características de los datos de entrada, que genera como salida el correspondiente *embedding*. Este se puede entender como un descriptor a alto nivel que representa la transformación de los datos de entrada en un nuevo espacio de características. Ambas ramas se unen mediante una función que utiliza una métrica determinada para calcular la distancia entre las características (esto es, las diferencias entre las codificaciones). Partiendo de este valor, se puede deducir el grado de similitud entre los datos de cada rama usando una función discriminante. La salida del sistema es un valor binario que representa la pertenencia o no de las entradas a un mismo grupo. Se trata, por tanto, no de un sistema que aprende las características para cada una de las clases, sino de un detector de similitudes capaz de reconocer a partir de éstas, si dos instancias corresponden a una misma categoría (*matching*) mediante una función de similitud.

El potencial de las RSs se destaca en estos 4 ámbitos:

- Disponibilidad de datos: el tamaño del *dataset* a analizar no es relevante para el proceso de *matching,* ya que el sistema no aprende cada clase en particular. La importancia reside en que el extractor de características sea adecuado para la aplicación dada.
- Generalización: en caso de ampliación o reducción del sistema mediante la introducción o eliminación de clases, no es necesario un reentrenamiento del mismo.
- Consistencia: la réplica de los pesos en cada red hace que unos datos de entrada determinados generen la misma codificación independientemente de la rama que los procese.
- Simetría: dados dos conjuntos de datos de entrada, el resultado de la clasificación generado por la métrica de comparación será el mismo aun cuando se inviertan sus ramas.

Para el entrenamiento de las RSs se utilizan duplas de imágenes con una etiqueta binaria que indica su pertenencia a la misma clase. La etiqueta y el valor predicho son los parámetros de entrada de una función de pérdida que se utiliza para actualizar los pesos de la red. Existe un tipo de entrenamiento más sofisticado, que requiere 3 instancias como entrada y no necesita etiquetas de forma explícita: el entrenamiento mediante *triplets*.

El método de entrenamiento mediante *triplets* [46] varía la estructura de la RS para modificar la forma de aprendizaje. Su arquitectura se compone en este caso de 3 redes iguales que comparten los pesos. Cada una de estas redes genera un *embedding* a partir de una de las siguientes entradas:

1- Ancla (A) recibe una clase cualquiera;

2- La entrada positiva (P) recibe otra instancia perteneciente a la misma clase que el ancla;

3- La entrada negativa (N) recibe una clase diferente.

La idea de este tipo de entrenamiento es acercar en el espacio de características objetivo la entrada positiva al ancla y separar o alejar la entrada negativa. Para ello, se calculan las distancias A-P y A-N mediante una función distancia d. Estos valores sirven como parámetros a la función de pérdida $\mathcal{L}$,

utilizada para actualizar los pesos de la red. De esta forma, el entrenamiento mediante *triplets* hace que el sistema sea capaz de aprender simultáneamente instancias positivas y negativas. Además, el gran número de combinaciones que se pueden seleccionar como *triplets* para un *dataset* reduce la influencia del sobreentrenamiento (*overfitting).*

Basados en cómo de diferentes son las instancias entre sí, los *triplets* se pueden clasificar en 3 categorías:

- *Easy triplets*: *triplets* que tienen una pérdida de cero.

$$d(A, P) + \alpha < d(A, N) \tag{1}$$

- *Hard triplets*: *triplets* en donde la entrada negativa está más cerca al ancla que la entrada positiva.

$$d(A, N) < d(A, P) \tag{2}$$

- *Semi-hard triplets*: *triplets* en los que la entrada negativa no está más cerca al ancla que la positiva, pero todavía existe pérdida positiva.

$$d(A, P) < d(A, N) < d(A, P) + \alpha \tag{3}$$

donde α es el margen que describe la distancia mínima que debe existir entre P y N.

Normalmente el conjunto de entrenamiento está compuesto por *triplets* de las 3 categorías con un porcentaje prefijado para cada una de ellas. Existen técnicas de minería de datos que, realizando previamente un preprocesado de los datos, se encargan de analizar el *dataset* para extraer *triplets* en función de su dificultad. Los preferidos para el entrenamiento son los *hard triplets,* ya que son los que aportan más información al sistema y más contribuyen a la convergencia durante el entrenamiento. Visualmente se entienden este tipo de *triplets* como el conjunto en donde las imágenes P y N son muy parecidas, pero pertenecen a clases diferentes (p.e. perro/lobo).

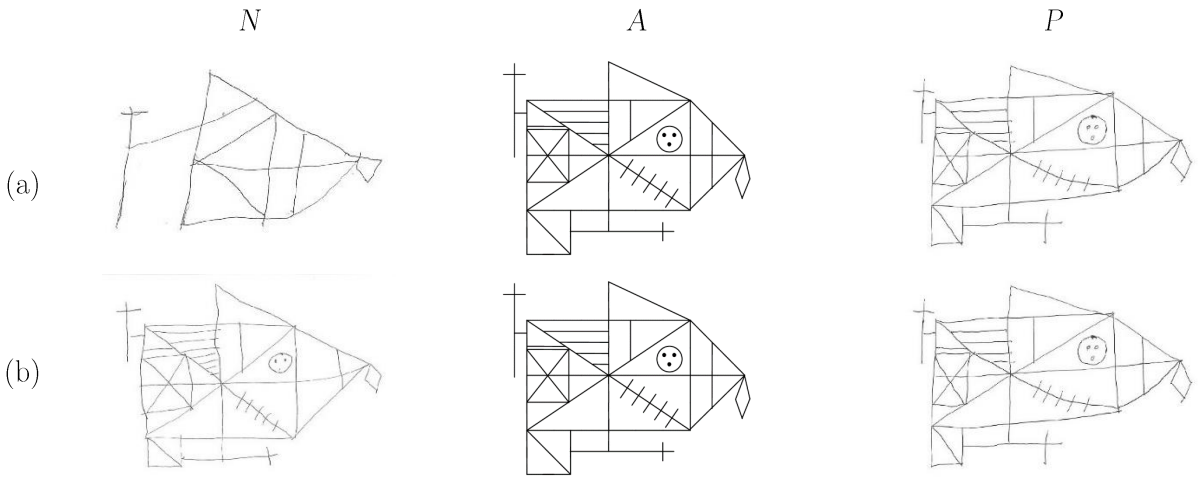


Figura 7. La fila (a) muestra un ejemplo de *easy-triplet* mientras que la fila (b) muestra una *hard-triplet.* Las figuras centrales representan A, en la derecha se representa P y en la izquierda N.

En el sistema propuesto, excluyendo las escasas instancias donde el deterioro cognitivo es avanzado y se puede observar que los dibujos difieren claramente del patrón (*easy triplets,* Figura 7.a), la categoría a la que pertenecen los *triplets* de entrenamiento corresponde a *hard triplets* (Figura 7.b). Este hecho simplifica el problema durante el entrenamiento, eliminando el paso de obtención de *triplets* preferidos

mediante minería de datos. Además, la instancia asignada como ancla se reduce únicamente a la FCR no modificada, ya que es el único modelo utilizado en dicha prueba. Esta condición simplifica y facilita de nuevo el entrenamiento y la evaluación, aliviando la complejidad en la selección de *triplets* y descartando que se puedan seleccionar posibles *outliers* como ancla, hecho que supondría un inconveniente para la generalización del sistema final.

3.3 Arquitecturas para el análisis de la prueba de la FCR

Este trabajo presenta 3 arquitecturas para la clasificación de copias de dibujos de la FCR. La primera red se basa en una RNA de clasificación binaria. La segunda utiliza una estructura de RS. La tercera modifica la anterior para su entrenamiento a través de *triplets*.

Como elemento común, todas las redes tienen un extractor de características preentrenado (Figura 8) basado en CNNs. Se han realizado experimentos con diferentes arquitecturas de la CNN utilizando el SQD para encontrar su estructura óptima. Estos experimentos se han ejecutado para diferentes valores de profundidad de la red y número de filtros. La CNN resultante se compone de 3 capas con etapas de filtrado (convolución), activación mediante ReLU (*Rectified Linear Unit*) y *pooling*. Como entrada recibe una imagen de tamaño 100x100 píxeles, a la que se le aplican 64 filtros en la primera capa y 32 en las restantes. Respecto a los filtros iniciales, existen trabajos que proponen tamaños de hasta 15x15 [47] que pretenden procesar más información del contexto en la primera convolución, y otros que deciden utilizar filtros muy pequeños 3x3 obviando esa información. Se ha optado por una solución intermedia con filtros 5x5 para la capa de entrada y 3x3 para las capas posteriores. Todas ellas realizan un *pooling* de 2x2. La salida del extractor se presenta como un vector de 6400 valores que representa el *embedding* del sistema. El tamaño elegido se sitúa entre los tamaños más grandes (~16.000) [48] encontrados en trabajos previos para codificar dibujos y las representaciones más compactas (64) [37].

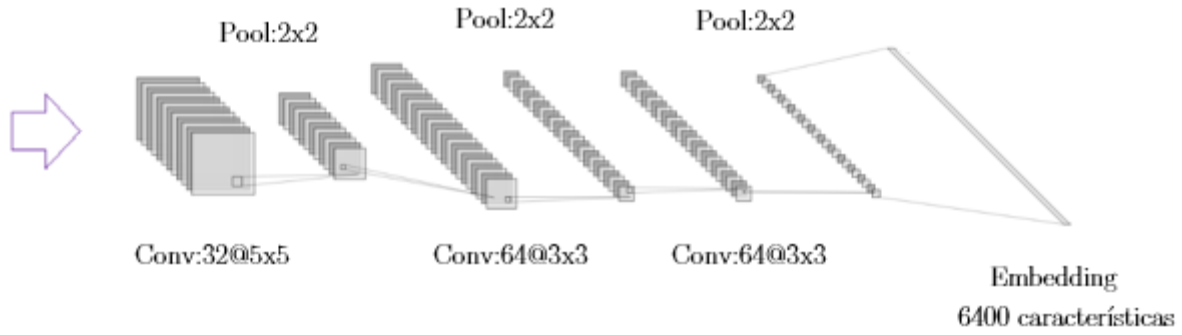


Figura 8. Arquitectura interna del extractor de características.

Para el entrenamiento del extractor con la técnica de TL, se ha añadido a continuación del *embedding* una RNA con una capa de salida con 19 neuronas activadas por *softmax* (una salida por clase). La RNA se compone de 2 capas intermedias FC (*fully-connected*), que tienen un tamaño de 128 neuronas con función de activación ReLU. La inicialización de los pesos se realiza mediante el inicializador Glorot [49] que genera una distribución normal centrada en 0 con varianza:

$$\forall i \qquad Var[W^i] = \frac{2}{n_i + n_{i+1}} \tag{4}$$

donde W^i son los pesos de la capa i, n_i el número de entradas de la capa y n_{i+1} el número de salidas. El algoritmo de optimización es el *Adam* [50], que es una extensión al descenso del gradiente estocástico. Debido a la exclusividad mutua de las clases, se utiliza la función de pérdida *sparse categorical crossenthropy*:

$$\mathcal{L} = -\sum_{i=1}^{k} y_i \cdot \log(y_i) \tag{5}$$

Las 3 arquitecturas utilizan el extractor preentrenado anteriormente con el SQD. Tanto para la arquitectura de RNA como para la RS entrenada con *triplets*, el entrenamiento consta de una segunda etapa en la que las redes se refinan con el DFCR. Para el caso de la RS, existe además un paso intermedio entre etapas, en el cual la red, ya con arquitectura de RS, se refina utilizando de nuevo el SQD. Las arquitecturas de cada una de las redes se presentan a continuación:

a) RNA: para la clasificación mediante la RNA se ha mantenido la estructura utilizada para el entrenamiento del extractor, sustituyendo la capa de salida *softmax* por una única neurona con activación sigmoide encargada de la clasificación binaria (Figura 9.a). La función de pérdida en este caso se define mediante la función *binary cross-entropy*;

$$\mathcal{L} = \ y \cdot \log y + (1-y) \cdot \log(1-y) \tag{6}$$

b) RS: la red se compone de dos ramas que corresponden al mismo extractor de características entrenado anteriormente (Figura 9.b). Siguiendo el estudio de [51], que presenta el impacto positivo de capas FC después de CNN en tareas de clasificación de imágenes, se ha añadido a cada una de las ramas una capa con 4096 neuronas activadas por la función sigmoide con una regularización de los pesos L2:

$$\lambda \sum_{i=0}^{N} W_i^2 \tag{7}$$

siendo λ el factor de regularización igual al valor típico 0'001. La función de distancia se calcula en forma de vector siguiendo la fórmula $\mathbf{w} = |\mathbf{x} - \mathbf{y}|$, siendo $\mathbf{x}$ e $\mathbf{y}$ los vectores que definen la codificación de las características para cada una de las ramas. La capa de salida se añade justo después del cálculo de la distancia entre vectores y está formada por una neurona activada por la función sigmoide. Como función de pérdida se toma *binary cross-entropy*.

c) RS entrenada mediante *triplets*: la estructura para el entrenamiento comprende 3 copias iguales del extractor previamente entrenado (Figura 9.c). A cada rama se le han añadido 2 capas de neuronas FC, con 4096 y 256 unidades respectivamente, para adecuar el tamaño de entrada de la función de pérdida a los valores típicos del entrenamiento con *triplets* [8]. Los pesos de estas capas utilizan una regularización L2 similar a la arquitectura anterior, y se inicializan siguiendo la distribución uniforme dentro del rango [-*límite*, *límite*], donde $límite = \sqrt{\frac{6}{n_i}}$ siendo n_i el número de pesos de entrada a una neurona i. A los valores de salida se les aplica la normalización L2 [8]. La función de pérdida es, en este caso, la función *triplet loss*:

$$\mathcal{L}(A, P, N) = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0\right) \tag{8}$$

donde f representa la función que genera el *embedding* y α es la distancia mínima deseada entre P y N. Para la fase de clasificación binaria, se ha aislado el extractor, congelado sus pesos y añadido una neurona de salida de activación sigmoide.

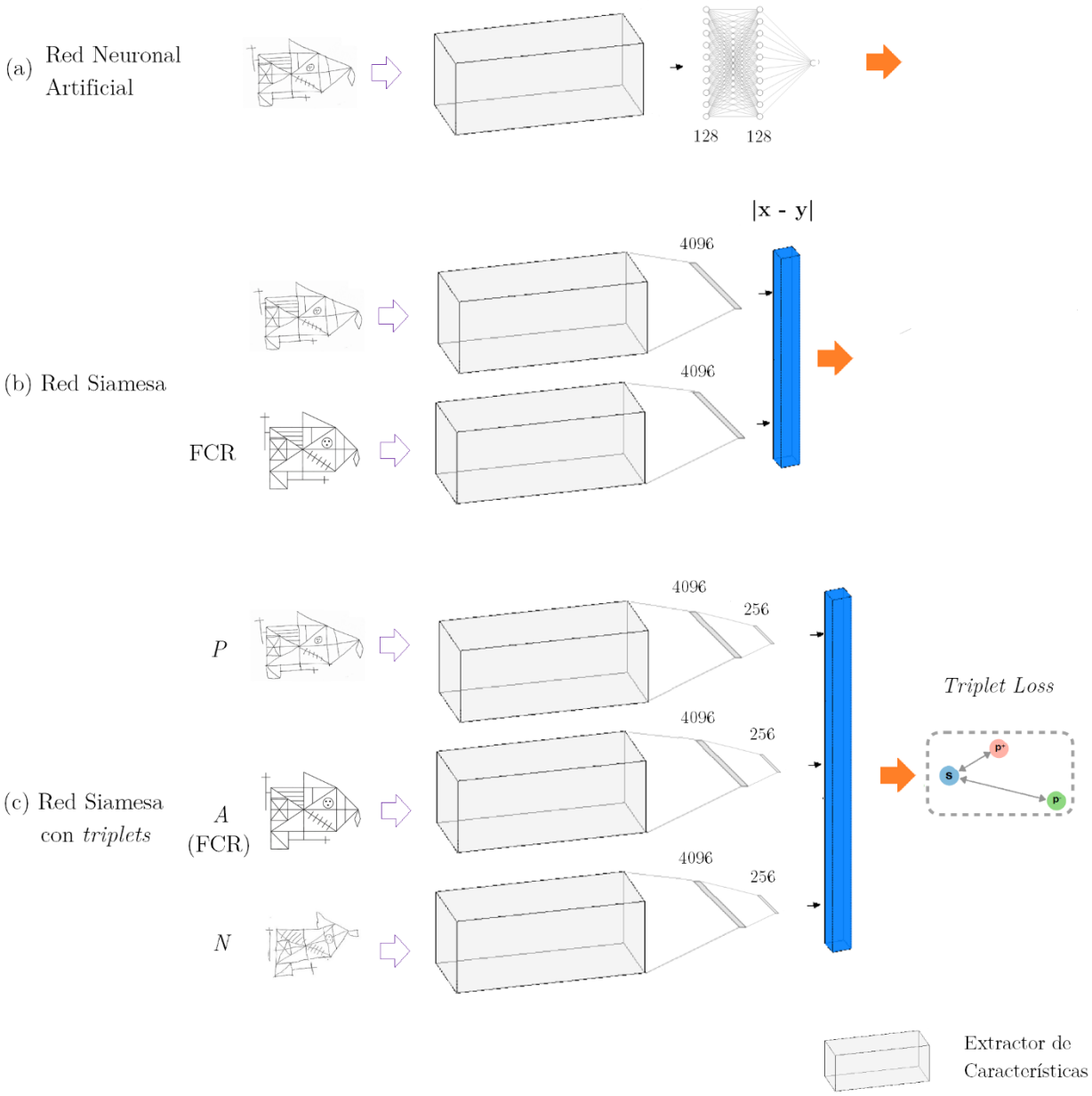


Figura 9. Representación de las 3 arquitecturas utilizadas en este trabajo. Se muestra la RNA (a), la RS (b) y su variante para el entrenamiento con *triplets* (c).

3.4 Interpretación de los resultados

Para poder interpretar los resultados de cada arquitectura, se utiliza la medida de sensibilidad, que indica la capacidad del sistema para identificar pacientes "no sanos", y de especificidad, que indica la capacidad para clasificar correctamente a pacientes "sanos". Estas medidas vienen definidas de la siguiente forma:

$$Sensibilidad = \frac{VP}{VP + FN} \tag{9}$$

$$Especificidad = \frac{VN}{VN + FP} \tag{10}$$

donde *VP*, *FN*, *VN* y *FP* significan verdadero positivo, falso negativo, verdadero negativo y falso positivo respectivamente.

4. Resultados experimentales

4.1 Experimentos para la configuración del extractor de características

Los primeros experimentos se han centrado en encontrar la configuración óptima de los parámetros para el extractor de características, por ser éste la pieza básica de las 3 arquitecturas. Se ha prestado especial atención al compromiso que debe existir entre la capacidad de generalización en la extracción de características y la especialización en las características propias de los dibujos del DFCR. Para ello, se han realizado experimentos enfocados a la configuración de parámetros como el tamaño de *batch,* el *dropout* o el *learning rate* (*lr*). Para este último, se ha identificado en primer lugar el *lr* óptimo para la fase de TL, para después entrenar cada arquitectura con el DFCR con un *lr* elegido de un subconjunto de valores siempre inferiores al anterior. De esta forma, se mantiene el conocimiento aprendido sobre las características generales de dibujos y se refina posteriormente.

Para el entrenamiento del extractor se ha dividido el SQD en 3 subconjuntos: el conjunto de entrenamiento con el 60% de instancias de cada clase, el conjunto de validación con un 20% y el conjunto de test con el 20% restante. Para la elección de parámetros se ha utilizado *cross-validation* con 4-*fold* y los experimentos han resultado en un entrenamiento final del extractor a lo largo de 20 *epochs*, con un tamaño de *batch* de 16 y un *lr* de 0'0015. La Tabla 3 muestra las precisiones de validación obtenidas después del entrenamiento con un tamaño de *batch* de 8, 16 y 32 y valores de *lr* en el rango [0'0008, 0'0019].

		Learning rate					
		0'0008	0'0009	0'001	0'0011	0'0012	0'0013
Tamaño de *batch*	8	66'14	66'12	67'2	67'28	70'05	71'81
	16	69'77	70'99	72'97	73'53	76'83	76'34
	32	75'67	75'79	75'85	77'62	77'59	79'43

		Learning rate					
		0'0014	0'0015	0'0016	0'0017	0'0018	0'0019
Tamaño de *batch*	8	74'95	75'25	76'4	76'69	76'65	76'88
	16	80'01	**80'72**	77'89	78'56	78'12	77'21
	32	80'22	80'28	79'74	79'89	78'33	78'23

Tabla 3. Precisión de validación de diferentes combinaciones de tamaño de *batch* y *lr* para la configuración del extractor de características.

Manteniendo la idea de generalización de las características del extractor, se han realizado experimentos aplicando la técnica de *dropout*. La Figura 10 muestra los resultados para el entrenamiento del extractor con el SQD con los parámetros definitivos. La precisión de entrenamiento sin la estrategia de *dropout*

alcanza un 99%, reduciéndose en un 5% para un *dropout* del 20%. Sin embargo, la diferencia entre las precisiones de validación se sitúa solo en un 0'1%. Este *dropout* del 20% se mantiene en los entrenamientos posteriores de las 3 arquitecturas.

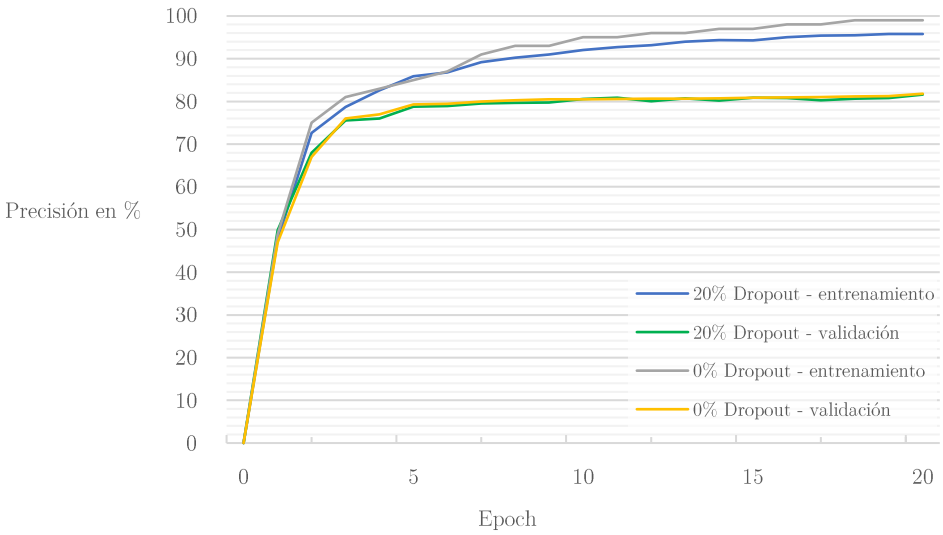


Figura 10. Precisiones de entrenamiento y validación del extractor para valores del 0 y 20% de *dropout* con los valores óptimos de *lr*, tamaño de *batch* y *epochs*.

4.2 Experimentos para la configuración de las distintas arquitecturas

Partiendo del extractor entrenado con los valores presentados en la sección anterior, se han realizado experimentos para hallar la configuración óptima de cada arquitectura. Los conjuntos de entrenamiento, validación y test en los que se ha dividido el DFCR contienen las mismas instancias para todas las arquitecturas y mantienen una proporción del 70%-15%-15% respectivamente. La configuración final para cada arquitectura se presenta a continuación:

a) RNA: para la elección de parámetros para la RNA con el DFCR, se ha utilizado *cross-validation* con 4-*fold*. Se ha seleccionado finalmente un tamaño de *batch* de 8 en 30 *epochs* para el entrenamiento. Para conservar el conocimiento aprendido en el TL, la selección del *lr* se ha realizado de un conjunto con valores siempre inferiores al *lr* del preentrenamiento del extractor, siendo el valor final 0'001.

b) RS: para plasmar la idea de compromiso entre generalización y especialización, se ha decidido en primer lugar, entrenar la red con el SQD para que se identifiquen las diferencias que existen entre los dibujos de *QuickDraw*. El tamaño del conjunto de entrenamiento, validación y test contienen duplas aleatorias de todas las clases con un tamaño de 5000, 512 y 512 respectivamente. El contenido de las duplas varía en cada *epoch*, garantizándose siempre la exclusividad entre conjuntos. La mitad de las parejas de cada conjunto contienen instancias de la misma clase, y la otra mitad de distinta.

Para limitar el aprendizaje en esta etapa, el *lr* inicial se modifica en cada *batch* siguiendo la función $lr_i = lr_{i-1} \cdot \frac{1}{(1+0\acute{}00025 \cdot i)}$, en donde i es el índice del *batch* en un *epoch* y lr_i el *learning rate* correspondiente para ese *batch*. Los parámetros elegidos a partir de los experimentos realizados para encontrar valores óptimos son un *lr* inicial de 0'001, un tamaño de *batch* de 16 y un entrenamiento durante 12 *epochs*. La Figura 11 muestra las diferencias encontradas durante los experimentos de las precisiones de validación para diferentes valores de *batch* en función de algunos de los valores de *lr*

utilizados en este paso.

Para el entrenamiento con el DFCR, se han creado duplas con todas las instancias del *dataset* siempre usando la FCR como uno de sus elementos. Para la generación de los conjuntos de entrenamiento, validación y test, el número de duplas positivas y las negativas ha mantenido la proporción entre instancias positivas y negativas del DFCR completo. Los valores óptimos encontrados corresponden con un *lr* igual a 0'0005, un tamaño de *batch* de 16 y un entrenamiento durante 50 *epochs*. El valor de los *lr* utilizados en los experimentos de esta fase también han sido siempre menores que los fijados en etapas anteriores.

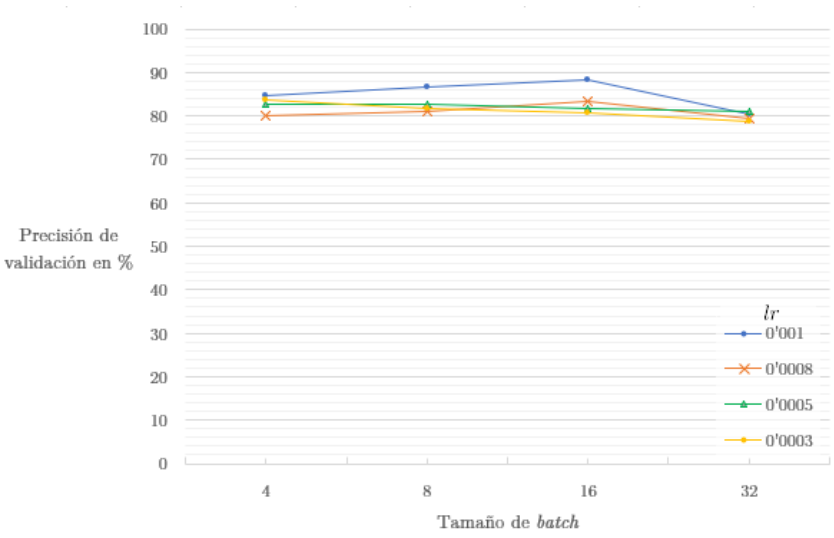


Figura 11. Precisión de validación con el SQD de la RS en función del tamaño del *batch* con diferentes *lr*.

c) RS entrenada mediante *triplets*: para el entrenamiento con el DFCR se han calculado todas las combinaciones de *triplets* posibles con el 70% de los datos, manteniendo la proporción de instancias positivas y negativas (26910 *triplets*, formadas a partir de 195 instancias positivas y 138 negativas), siendo la FCR siempre el ancla. El conjunto de validación y test contienen cada uno las combinaciones de un 15% de instancias del DFCR (1260 *triplets,* 42 positivas y 30 negativas). Los valores seleccionados para el entrenamiento mediante *triplets* han sido fijados en un *lr* de 0'0003 y un tamaño de *batch* de 8 durante 3 *epochs* con un valor de $\alpha = 0'2$ para la función de pérdida. Por otra parte, los experimentos para el entrenamiento del clasificador final se han realizado con el conjunto de test anterior, con un resultado final de un *lr* de 0'0005 durante 100 *epochs*.

4.3 Análisis del rendimiento de las distintas arquitecturas

Esta sección presenta los resultados obtenidos en los experimentos para cada arquitectura según los *datasets* con los que ha sido entrenada. La Tabla 4 muestra por filas estos resultados para cada conjunto (entrenamiento, validación y test):

a) RNA + SQD: la RNA ha sido entrenada con el SQD y presenta los resultados de la clasificación para las clases del SQD. Representa la configuración utilizada para el entrenamiento del extractor de características.

b) RNA + SQD + DFCR: a partir del entrenamiento de toda la red en a), la RNA ha sido refinada con el DFCR y presenta los resultados de la clasificación binaria con el DFCR.

c) RS + SQD: utiliza el extractor entrenado en a) y se refina entrenando la red de nuevo con SQD para la tarea de clasificación de las correspondientes clases.

d) RS + SQD + DFCR: a partir del estado del sistema alcanzado en c), se realiza un refinamiento con DFCR para la clasificación binaria.

e) RS con *triplets* + DFCR: utiliza el extractor entrenado en a) y se refina con el *dataset* DFCR. Los resultados corresponden a la clasificación con la modificación en la arquitectura explicada en secciones anteriores.

Debido al reducido tamaño de los conjuntos de entrenamiento y validación para el DFCR, cada instancia correctamente clasificada supone una mejora del 1'3$\bar{8}$% en los resultados obtenidos para este *dataset*, por lo que, para poder profundizar en su significado, se desglosan en forma de matrices de confusión (Tabla 5). La leve variación (~3%) entre los resultados en la validación y el test en todos los experimentos indican una distribución similar de los correspondientes conjuntos seleccionados.

Configuración	Entrenamiento	Validación	Test
a) RNA + SQD	95'75	80'72	**79'64**
b) RNA + SQD + DFCR	60'74	58,33	**56'94**
c) RS + SQD	98'29	88'2	**85'74**
d) RS + SQD + DFCR	98'76	68'05	**66'66**
e) RS *triplets* + DFCR	87'19	84'72	**86'11**

Tabla 4. Precisión obtenida para cada arquitectura en función de los *datasets* con los que han sido entrenadas.

		Real	
		No sano	Sano
Predicción	No sano	30	31
	Sano	0	11

(a) RNA

		Real	
		No sano	Sano
Predicción	No sano	13	7
	Sano	17	35

(b) RS

		Real	
		No sano	Sano
Predicción	No sano	24	4
	Sano	6	38

(c) RS con *triplets*

Tabla 5. Matrices de confusión para las arquitecturas RNA (a), RS (b) y RS con *triplets* (c).

5. Discusión

En primer lugar, se debe comentar la importancia de los experimentos que han ayudado durante la configuración del extractor de características y las distintas arquitecturas. En el caso del *dropout* (Figura 10), la pequeña diferencia en la precisión de validación entre los dos valores y la diferencia del 5% en la precisión de entrenamiento, sugiere que se ha alcanzado una mayor generalización del sistema con un *dropout* igual al 20%. Durante los experimentos de refinado de los extractores mediante el tamaño de *batch* en la RS (Figura 11), se ha observado que para valores inferiores a 16 con el valor

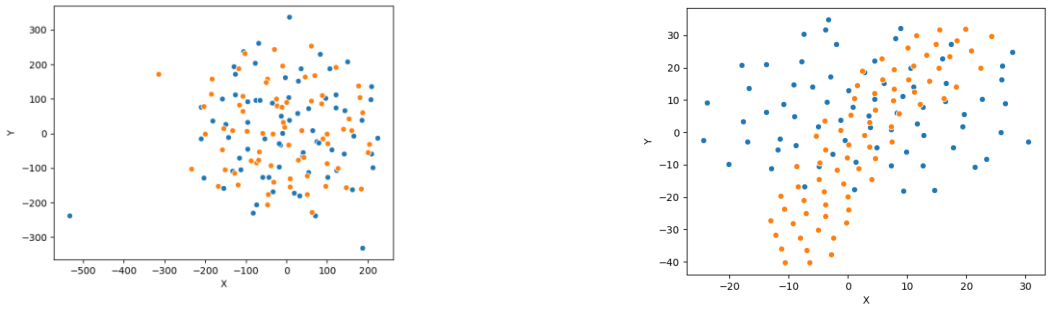
óptimo de *lr* de 0'001, la convergencia es más rápida durante el entrenamiento, sin embargo, la precisión decae debido al *overfitting*. En el caso de tamaños más grandes, se ha observado una reducción de la precisión final, propiciada por una pobre generalización del sistema. En las arquitecturas de la RNA y de la RS, los resultados reflejan una diferencia de precisión considerable en función del *dataset* utilizado en la clasificación (Tabla 4). La precisión de clasificación para el SQD supera a la obtenida con el DFCR en casi un 23% para la RNA y un 19% para la RS. Por ello, se podría contemplar la opción de modificar la configuración de parámetros que faciliten el aprendizaje después del TL, como puede ser el aumento del *lr* o realizar un refinado con un DFCR ampliado (mayor número de instancias).

Los resultados obtenidos en la precisión de test para la RNA con el DFCR superan en solo un 6% a los resultados que generaría un clasificador aleatorio. La influencia que podría tener el uso de la FCR como imagen de referencia en el entrenamiento no se puede aplicar, y se reduce a ser una instancia positiva cualquiera en el proceso. El sistema tiene una sensibilidad del 100% (Tabla 5 (a)), que contrasta con una especifidad del 26%. La baja especifidad supone una alta probabilidad de clasificar incorrectamente a pacientes sanos, con la consecuente pérdida de utilidad de la aplicación para el diagnóstico. El problema contrario se presenta para la RS, donde la sensibilidad solo alcanza un 43% y la especificidad se eleva a un 83% (Tabla 5 (b)). En este caso, existe una alta probabilidad de que un paciente con signos de deterioro cognitivo fuese considerado sano, por lo que sería recomendable utilizar otros métodos de diagnóstico como apoyo al diagnóstico final. A pesar de estos indicadores, el cambio de paradigma a *one-shot learning*, mejora los resultados globales anteriores, alcanzando un 66% de precisión (Tabla 4). Esta mejora refuerza el planteamiento de contemplar el problema como detección de similitudes entre dos imágenes, la del paciente y la FCR. La inclusión de la técnica de entrenamiento con *triplets* mejora considerablemente los resultados anteriores. La separación de instancias positivas y negativas a partir de la imagen modelo crea un espacio de características capaz de clasificar a los pacientes con una precisión del 86'11%. Del mismo modo, los porcentajes de sensibilidad y especificidad se sitúan en un 80% y un 90'4% respectivamente (Tabla 5 (c)). Valores tan altos en ambos indicadores son siempre deseables en pruebas diagnósticas para poder identificar el mayor número de casos de personas enfermas (sensibilidad) y confirmar los casos descartando sujetos sanos (especificidad).

La técnica de entrenamiento mediante *triplets* produce la separación de instancias positivas y negativas respecto al patrón. Para visualizar los espacios de salida generados, se ha utilizado la técnica de reducción de dimensionalidad *t-distributed stochastic neighbor embedding* (t-SNE) para proyectar las instancias del conjunto de test en 2 dimensiones. La Figura 12 (a) muestra el estado inicial antes de comenzar el entrenamiento con el DFCR. El entrenamiento mediante *triplets* separa para cada actualización de los pesos la representación de las imágenes de los no sanos respecto al patrón, al mismo tiempo que acerca a los sanos. La Figura 12 (b), (c) y (d) muestran una proyección en 2D del estado del sistema en cada *epoch*. Aunque se puede intuir previamente, la Figura 12 (d) representa cómo los datos se agrupan en dos *clusters* diferentes.

Siguiendo con la arquitectura de RS entrenada mediante *triplets,* se presentan algunas de las instancias del DFCR clasificadas de forma errónea. La Figura 13 (a) y (b) muestran dos dibujos de personas sanas clasificadas como no sanas. El paciente (a) obtuvo una puntuación perfecta (36) en la prueba real y la puntuación otorgada por el experto en el caso (b) es 19, por lo que la persona también fue clasificada como no sana. La Figura 13 (c) y (d) presentan dos pacientes no sanos clasificados como sanos. La puntuación real de la prueba fue en este caso 35 (diagnóstico como sano) y 19 respectivamente. Se puede ver que existen casos en los que un diagnóstico erróneo coincide con el diagnóstico real del experto. Recordemos que el diagnóstico del sujeto se basa en un conjunto de pruebas diagnósticas, no solo en el resultado de la prueba de la FCR, por lo que puede haber casos en los que los resultados no

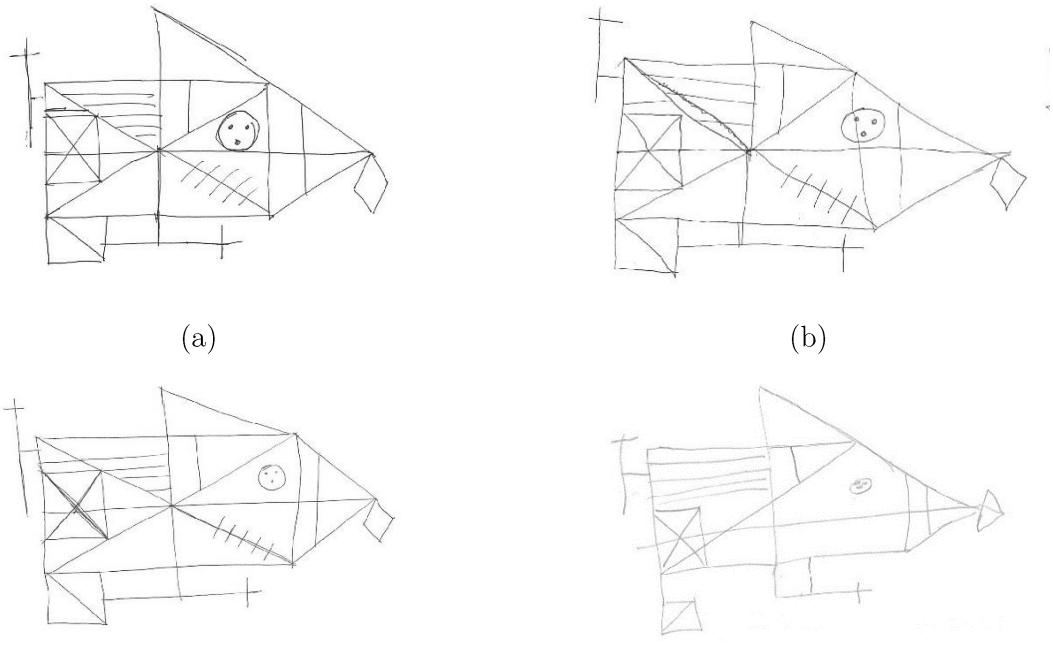
coincidan.



(a) Epoch 0 (b) Epoch 1

(c) Epoch 2 (d) Epoch 3

Figura 12. Representación en un espacio vectorial de 2 dimensiones mediante t-SNE de las salidas generadas por la RS entrenada con *triplets* para el caso inicial (a), para el *epoch* 1 (b), para el *epoch* 2 (c) y para el *epoch* 3 (d).



(a) (b)

(c) (d)

Figura 13. Ejemplos de Falsos Positivos (a) y (b) y de Falsos Negativos (c) y (d) para la RS entrenada con *triplets*.

6. Análisis ético y social

El tema de este trabajo se ha centrado en el diagnóstico de enfermedades neurodegenerativas, las cuales afectan principalmente a personas de avanzada edad. Este grupo es especialmente sensible en temas relacionados con la salud, movilidad y bienestar. Es por ello que los nuevos avances tecnológicos deben ser capaces de adaptarse a estas circunstancias para satisfacer las necesidades especiales que se desprenden de esta realidad.

No hay que perder de vista que el objetivo último de la prueba de la FCR es ayudar a diagnosticar lo antes posible y de manera eficaz este tipo de enfermedades. Si bien un diagnóstico temprano junto con el correspondiente tratamiento no elimina la enfermedad, puede mitigar sus efectos, mejorar la calidad de vida en la etapa final y favorecer el entendimiento del problema por parte del entorno familiar. Es por esto que las aplicaciones de diagnóstico basadas en Inteligencia Artificial (IA) tienen que jugar un papel importante independientemente de la dificultad que represente el problema.

El objetivo de la automatización es la liberar al ser humano del trabajo que puede ser realizado por una máquina. Las aplicaciones de diagnóstico automático que utilizan IA tienen que contribuir también a alcanzar este objetivo y, de forma ideal, a la mejora de los resultados de diagnóstico que puede proporcionar el personal médico actualmente. Las consecuencias más inmediatas de la implantación del sistema presentado en este trabajo significarían una reducción del tiempo de análisis de dibujos por parte del personal y de los recursos necesarios para la prueba de la FCR. Esto supone la posibilidad de dedicar este tiempo, dinero y material a otras tareas en las que no sea posible la automatización. Para el caso de este tipo de enfermedades, la importancia de la automatización radica además en la influencia del rango de edad, que hace que el grupo objetivo sea cada vez más numeroso. Una automatización permitiría incluso implantar estrategias de *screening* (pruebas masivas enfocadas a ciertos grupos de la sociedad sin síntomas), de forma que todas las personas en un rango de edad pudiesen acceder a un diagnóstico de enfermedades neurodegenerativas de forma sencilla y cómoda. Teniendo en cuenta la posible limitación de movilidad, se pueden plantear soluciones como el envío a casa o a centros de ancianos de la prueba de la FCR con las correspondientes instrucciones de ejecución y devolución al centro médico. Los resultados de esta prueba se podrían derivar entonces a los centros de atención primaria y junto con el resto de pruebas de la batería correspondiente, generar el diagnóstico final. No hay que olvidar la compatibilidad entre diferentes pruebas de diagnóstico. El entrenamiento de este sistema automático podría estar enfocado a alcanzar una alta sensibilidad (o especificidad) para complementarse con otras pruebas diagnósticas con un alto grado de especificidad (o sensibilidad), o incluso con el mismo sistema entrenado de manera diferente. De esta forma se puede conseguir un sistema de diagnóstico robusto, eficaz y sencillo.

Desde el punto de vista legal, la relación entre IA y medicina se mueve, al igual que en la mayoría de campos, en un marco regulatorio difuso. Aunque inicialmente la responsabilidad recae en el médico encargado del diagnóstico, con la introducción de sistemas automáticos se debe abordar una posible adaptación en la reglamentación. Se deben fijar procedimientos claros para la aceptación e implantación de sistemas inteligentes en centros médicos, del mismo modo que se deben establecer criterios mínimos de calidad en función de los resultados y la seguridad. No es una tarea sencilla debido a los diferentes niveles de organización que lo gestionan (local, autonómico, nacional y europeo para el caso de España), así como el tipo de entidades involucradas (públicas/privadas). A esto hay que añadir la preferencia que puede existir por parte de ciertas personas u organizaciones para elegir un conjunto de pruebas distinto a las que incluyen la prueba de FCR (para enfermedades de carácter neurodegenerativo).

Independientemente de factores laborales, económicos o materiales se debe tener claro que la IA tiene que servir como herramienta para el bienestar de las personas, especialmente de los más vulnerables, porque cabe recordar que, solo una sociedad que respeta a sus niños y mayores es digna de ser calificada como civilizada.

7. Conclusiones y trabajo futuro

En este trabajo se ha introducido la utilización de RSs para el diagnóstico de enfermedades cognitivas mediante pruebas médicas basadas en dibujos. De forma teórica, se ha razonado la adecuación del uso de las RSs para esta tarea. De forma experimental, se han presentado 3 arquitecturas diferentes, comparándolas y destacando los resultados de la RS entrenada con *triplets*, con la cual se han alcanzado valores de precisión del 86'11%. Para abordar el problema del tamaño del *dataset*, se ha propuesto el preentrenamiento de las redes utilizando un *subset* del *dataset QuickDraw*, cuyas instancias comparten características con los dibujos a analizar. Por todo ello, las RSs se consideran una opción adecuada como ayuda para el diagnóstico automático de la prueba de la FCR.

Los resultados obtenidos en este trabajo abren una línea de investigación para la utilización del *one-shot recognition* en este tipo de aplicaciones. Las posibilidades que ofrecen las arquitecturas de RSs son enormes y la construcción de un sistema más robusto puede venir de la mano de la aplicación de CNNs profundas, otro tipo de funciones de pérdida y distancia o mediante la modificación de las RNAs situadas entre el extractor y la función de distancia. En cualquier caso, se abre la puerta a una nueva herramienta para el análisis de dibujos en el ámbito del diagnóstico médico.

8. Referencias

[1] A. Rey, «L'examen psychologique dans les cas d'encéphalopathie traumatique. (Les problems.) [The psychological examination in cases of traumatic encepholopathy. Problems],» *Archives de Psychologie,* vol. 28, pp. 215-285, 1941.

[2] P. Osterrieth, «Le test de copie d'une figure complexe; contribution à l'étude de la perception et de la mémoire [Test of copying a complex figure; contribution to the study of perception and memory],» *Archives de Psychologie,* vol. 30, pp. 206-356, 1944.

[3] M. D. Lezak, Neuropsychological Assessment (2nd ed.), New York: Oxford University Press, 1995.

[4] R. A. Stern, E. A. Singer, L. M. Duke, N. G. Singer, C. E. Morey, E. W. Daughtrey y E. Kaplan, «The Boston qualitative scoring system for the Rey-Osterrieth complex figure: Description and interrater reliability,» *Clinical Neuropsychologist,* vol. 8, nº 3, pp. 309-322, 1994.

[5] J. Patterson y A. Gibson, Deep Learning: A practitioner´s approach, O´Really, 2017.

[6] P. Sangkloy, N. Burnell, C. Ham y J. Hays, «The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies,» *ACM Transactions on Graphics (proceedings of SIGGRAPH),* vol. 35, pp. 1-12, 2016.

[7] X. Wang y X. Tang, «Face Photo-Sketch Synthesis and Recognition,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, nº 11, pp. 1955-1967, 2009.

[8] Q. Yu, F. Liu, Y.-Z. SonG, T. Xiang, T. Hospedales y C. C. Loy, «Sketch Me That Shoe,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016),* pp. 799-807, 2016.

[9] L. Liu, F. Shen, Y. Shen, X. Liu y L. Shao, «Deep Sketch Hashing: Fast Free-hand Sketch-Based Image Retrieval,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017),* pp. 2298-2307, 2017.

[10] B. Lake, R. Salakhutdinov y J. Tenenbaum, «Human-level concept learning through probabilistic program induction,» *Science,* vol. 350, nº 6266, pp. 1332-1338, 2015.

[11] Y. LeCun, C. Cortes y C. Burges, MNIST Dataset, «http://yann.lecun.com/exdb/mnist/».

[12] M. Eitz, J. Hays y M. Alexa, «How Do Humans Sketch Objects?,» *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012),* vol. 31, nº 4, 2012.

[13] Google-Creative-Labs, QuickDraw Dataset, «https://github.com/googlecreativelab/quickdraw-dataset».

[14] S. Chopra, R. Hadsell y Y. LeCun, «Learning a similarity metric discriminatively, with application to face verification,» *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* vol. 1, pp. 539-546, 2005.

[15] F.-F. Li, R. Fergus y P. Perona, «A Bayesian approach to unsupervised one-shot learning of object categories,» *Ninth IEEE International Conference on Computer Vision,* vol. 2, pp. 1134-1141, 2003.

[16] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard y L. Jackel, «Backpropagation Applied to Handwritten Zip Code Recognition,» *Neural Computation,* vol. 1, nº 4, pp. 541-551, 1989.

[17] Y. LeCun, P. Haffner, L. Bottou y Y. Bengio, «Gradient-based learning applied to document recognition,» *Proceedings of the IEEE,* vol. 86, nº 11, pp. 2278-2324, 1998.

[18] O. Marbán, «CRM in e-Business: a Client's Life Cycle Model Based on a Neural Network,» de *E-Commerce and Intelligent Methods*, Heidelberg, Springer, 2002.

[19] D. Prokhorov, Computational Intelligence in Automotive Applications. Studies in Computational Intelligence, vol 132, Berlin: Springer, 2008.

[20] E. Pasero y W. Moniaci, «Artificial neural networks for meteorological nowcast,» *Proceedings of CIMSA 2004, IEEE Conference on Computer Intelligence for Measurement Systems and Applications,* pp. 36-39, 2004.

[21] S. Graham, E. Lee, D. Jeste, R. Van Patten, E. Twamley, C. Nebeker, Y. Yamada, H.-C. Kim y C. Depp, «Artificial intelligence approaches to predicting and detecting cognitive decline in older adults: A conceptual review,» *Psychiatry Research,* vol. 284, 2020.

[22] J. Bernal, K. Kushibar, D. Asfaw, S. Valverde, O. Arnau, R. Martí y X. Lladó, «Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review,» *Artificial Intelligence in Medicine,* vol. 95, pp. 64-81, 2019.

[23] B. Pardamean, T. W. Cenggoro, R. Rahutomo, A. Budiarto y E. Karuppiah, «Transfer Learning from Chest X-Ray Pre-trained Convolutional Neural Network for Learning Mammogram Data,» *Procedia Computer Science,* vol. 135, pp. 400-407, 2018.

[24] F. K. Alsheref y W. H. Gomaa, «Blood Diseases Detection using Classical Machine Learning Algorithms,» *International Journal of Advanced Computer Science and Applications,* vol. 10, nº 7, 2019.

[25] M. J. Kang, S. Y. Kim y D. Na, «Prediction of cognitive impairment via deep learning trained with multi-center neuropsychological test data,» *BMC Medical Informatics Decision Making,* vol. 19, 2019.

[26] F. Bertè, G. Lamponi, R. S. Calabrò y P. Bramanti, «Elman neural network for the early identification of cognitive impairment in Alzheimer's disease,» *Functional Neurology,* vol. 29, pp. 57-65, 2014.

[27] H. T. Gorji y N. Kaabouch, «A Deep Learning approach for Diagnosis of Mild Cognitive Impairment Based on MRI Images,» *Brain Sciences,* vol. 9, nº 217, 2019.

[28] O. Vinyals, C. Blundell, T. Lillicrap y D. Wierstra, «Matching networks for one-shot learning,» *Neural Information Processing Systems (NIPS),* 2016.

[29] B. Hariharan y R. Girshick, «Low-shot visual object recognition,» Facebook AI Research (FAIR), 2016.

[30] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger y R. Shah, «Signature verification using a "Siamese" time delay neural network,» *Advances in neural information processing systems,* vol. 6, pp. 737-744, 1993.

[31] R. Pflugfelder, «An In-Depth Analysis of Visual Tracking with Siamese Neural Networks,» University of Technology, Vienna, 2017.

[32] H. Wu, Z. Xu, J. Zhang, W. Yan y X. Ma, «Face recognition based on convolution siamese networks,» *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI),* pp. 1-5, 2017.

[33] M. Amin-Naji, H. Mahdavinataj y A. Aghagolzadeh, «Alzheimer's disease diagnosis from structural MRI using Siamese convolutional neural network,» *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA),* pp. 75-79, 2019.

[34] M. Li, K. Chang y B. Bearce, «Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging,» *npj Digital Medicine,* vol. 3, nº 48, 2020.

[35] Y.-A. Chung y W.-H. Weng, «Learning Deep Representations of Medical Images using Siamese CNNs with Application to Content-Based Image Retrieval,» Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, 2017.

[36] G. Koch, «Siamese neural networks for one-shot image recognition,» *International Conference on Machine Learning (ICML) Deep Learning Workshop,* vol. 2, 2015.

[37] Y. Qi, Y.-Z. Song, H. Zhang y J. Liu, «Sketch-Based Image Retrieval Via Siamese Convolutional Neuronal Network,» *2016 IEEE International Conference on Image Processing (ICIP),* pp. 2460-2464, 2016.

[38] F. Wang, L. Kang y Y. Li, «Sketch-based 3D Shape Retrieval using Convolutional Neural Networks,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015),* pp. 1875-1883, 2015.

[39] P. Lu, H. Lin, Y. Fu, S. Gong, Y.-G. Jiang y X. Xue, «Instance-level Sketch-based Retrieval by Deep Triplet Classification,» Fudan University, 2018.

[40] B. Zhu y E. Quigley, «Sketch-based Object Recognition,» Stanford University, 2014.

[41] S. Parui y A. Mittal, «Sketch-based Image Retrieval from Millions of Images under Rotation, Translation and Scale Variations,» Indian Institute of Technology, Madras, 2015.

[42] W. Lu y E. Tran, «Free-hand Sketch Recognition Classification,» Stanford University, 2017.

[43] M. Rincón, «An open datset for automatic drawing analysis of figures included in neuropsychological tests for assessment and diagnosis of mild cognitive impairment,» 2020.

[44] S. García Herranz, M. C. Díaz Mardomingo y H. Peraita, «Neuropsychological predictors of conversion to probable Alzheimer disease in elderly with mild cognitive impairment,» *Journal of neuropsychology,* vol. 10, nº 2, pp. 239-255, 2016.

[45] M. J. Benedet y M. Á. Alejandre, TAVEC. Test de Aprendizaje Verbal España-Complutense, Madrid: TEA Ediciones, 2014.

[46] F. Schroff, D. Kalenichenko y J. Philbin, «FaceNet: A unified embedding for face recognition and clustering,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015),* pp. 815-823, 2015.

[47] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang y T. Hospedales, «Sketch-a-Net: A Deep Neural Network that Beats Humans,» *International Journal of Computer Vision,* 2017.

[48] F. Zhu, J. Xie y Y. Fang, «Learning Cross-Domain Neural Networks for Sketch-Based 3D Shape Retrieval,» *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence,* p. 3683–3689, 2016.

[49] X. Glorot y Y. Bengio, «Understanding the difficulty of training deep feedforward neural networks,» *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics,* vol. 9, pp. 249-256, 2010.

[50] D. Kingma y J. Ba, «Adam: A Method for Stochastic Optimization,» *International Conference on Learning Representations,* 2014.

[51] B. Shabeer, S. R. Dubey, V. Pulabaigari y S. Mukherjee, «Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification,» *Neurocomputing,* vol. 378, pp. 112-119, 2020.