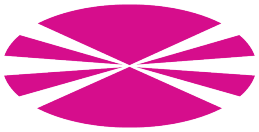

UNIVERSIDADE DA CORUÑA

Sistema inteligente para la detección del número de dedos levantados de una mano

Memoria Práctica Aprendizaje Automático

Fecha de inicio: 15/02/2024
Versión: 0.0

by

Lucía Álvarez García
l.alvarezg@udc.es
Simón Noya Domínguez
simon.noyad@udc.es
Samuel Pérez Fente
samuel.perez.fente@usc.es
David Naya Diaz
david.nayad@udc.es
Antón López Núñez
anton.lopez.nunez@udc.es

Universidade de A Coruña

Contents

1 Introducción

La capacidad de reconocer y comprender la posición de unas manos en imágenes es fundamental en aplicaciones de visión por computadora, desde interfaces hombre-máquina hasta robótica y seguridad. Dentro de este contexto, el reconocimiento preciso del número de dedos levantados en una mano humana es un desafío importante debido a la variabilidad en la forma, tamaño y orientación de las manos dentro de una imagen, así como las diferentes condiciones de iluminación.

En esta memoria nos enfocamos en el desarrollo de un sistema de reconocimiento de número de dedos levantados en imágenes de manos humanas (izquierda o derecha) mediante el uso de RR.NN.AA., SVM, kNN, y árboles de decisión; asimismo evaluamos también su rendimiento en conjuntos de datos estándar, y discutimos posibles aplicaciones y mejoras futuras. Este sistema tiene aplicaciones potenciales en campos como la medicina o los videojuegos (realidad virtual), entre otros. Para implementarlo hemos utilizado el lenguaje de programación Julia.

Este trabajo puede ser útil como prototipo para la identificación de objetos en una imagen, con futuras aplicaciones como reconocimiento de caras (entornos de seguridad, búsqueda de personas...), reconocimiento de gestos (traducción de lenguaje de signos, manos libres, videojuegos, accesibilidad...), etc. Hoy en día, sea en el mundo real o en el virtual, la accesibilidad tiene gran impacto e importancia, pero desgraciadamente es tratada como una "norma que hay que cumplir" en lugar de un objetivo relevante a perseguir y perfeccionar. El número de personas con diversas necesidades de adaptación no para de crecer, y es imprescindible enfocarse a una audiencia lo más amplia posible, desarrollando sistemas que puedan ser usados tanto por individuos sanos como por personas con discapacidades mentales o físicas (por ejemplo, reconocimiento de lenguaje de signos para sordos), sin olvidar los que requieren ajustes ligeros por problemas como falta de atención (importante incorporar el reconocimiento de gestos en entornos como la conducción que analicen la actividad del conductor para ayudar a evitar accidentes) o disautonomía (desmayos repentinos, importantes de identificar si el individuo en cuestión no está acompañado, ya que sería incapaz de pedir ayuda por sí mismo).

A continuación describiremos el problema en mayor detalle (consultar sección 2: Descripción del problema), haremos un análisis bibliográfico (consultar sección 3: Análisis Bibliográfico) de algunos de los trabajos más recientes y explicaremos todo el proceso de desarrollo del sistema (sección 4: Desarrollo) y sus resultados, junto con las conclusiones que deducimos de ellos (consultar sección 5: Conclusiones). También incluimos un apartado con una serie de líneas de trabajo que se podrían iniciar a partir de este sistema (consultar sección 6: Trabajo Futuro).

2 Descripción del problema

El problema a resolver con nuestro sistema es la detección del número de dedos levantados de una mano humana. Se toman imágenes de manos (izquierda o derecha) como entrada, y como salida se devuelve el número de dedos levantados (0, 1, 2, 3, 4, o 5).

Cada instancia de la base de datos consiste en una imagen cuyo nombre de archivo acaba en el nº de dedos levantados seguido de L ("left"), si es una mano izquierda, o R ("right"), si es una mano derecha. A continuación se muestra un ejemplo de las instancias utilizadas:

Figure 1: Mano izquierda con 3 dedos levantados

Figure 2: Mano izquierda con 4 dedos levantados

Figure 3: Mano izquierda con 5 dedos levantados

Todas las imágenes tienen una **resolución de 128x128px** y están en **escala de grises**. Todas las manos tienen la **palma de cara a la cámara** y, en caso de que un dedo esté levantado, lo estará completamente. Los dedos levantados apuntarán hacia arriba, y no les faltará ninguna articulación. Aunque los números 2 y 3 suelan ser mostrados de maneras diferentes dependiendo de los hábitos de la persona que gesticula, para este sistema siempre usaremos los dedos índice y medio para indicar el número 2, y el índice, medio y anular para el número 3 (como se puede ver en la figura 1). Los números 1, 4 (figura 2), 5 (figura 3) y el puño (nº 0) se representan de la manera más común.

La base de datos ha sido extraída de Internet, concretamente de "kaggle.com" (Koryakin (2018)). Su autor la ha elaborado tomando un vídeo de su mano derecha ante un fondo negro. Las **imágenes de la mano izquierda** las generó **invirtiendo las imágenes fuente** como un espejo. El preprocesado incluyó aplicar un filtro de escala de grises, incrementar el contraste, y recortar y centrar la imagen según su centro de masa. La base de datos cuenta con un **conjunto de entrenamiento de 18.000 imágenes** (**3.000 imágenes por cada número de dedos**, **1.500 de la mano derecha** y **1.500 de la mano izquierda**) y un **conjunto de test de 3.600**.

Puesto que este es un **problema de clasificación**, **no es necesario normalizar los valores de salida**. La **métrica utilizada** para evaluar el trabajo del modelo **es la precisión**, ya que la **base de datos está balanceada** y **todas las clases poseen el mismo nivel de importancia**.

3 Análisis Bibliográfico

Algunos trabajos relacionados con nuestro problema lo abordan en 2 pasos, como es el caso del trabajo de extracción de características de manos de Feng et al. (2011), en el que separan la fase de localización gruesa (CLP), donde el contorno de una mano es delimitado con un polígono usando líneas y puntos para que luego un algoritmo extraiga las características que interesan de la mano (como las puntas de los dedos, las articulaciones...); de la fase de localización refinada (RLP), donde se propone un algoritmo de refinamiento que trataría los datos anteriores. En este caso, el sistema se aplicó con éxito a un sistema de seguimiento de manos en 3D y un sistema de reconocimiento de formas de manos en 2D. Otros artículos como el de Chaudhary et al. (2013) simplemente analizan trabajos en el área de reconocimiento de gestos, afirmando que en general se centran en aproximaciones inteligentes incluyendo métodos computacionales como RR.NN.AA., fuzzy logic, algoritmos genéticos... y preprocesado de imágenes usando las puntas de los dedos para detección de manos.

En otros trabajos como el de Martín Rivas (2018) se tiene nuestro mismo objetivo, contar el número de dedos que tiene levantada una mano. Para ello toman las imágenes con una cámara infrarroja y luego utilizan cuatro métodos distintos para la identificación de las imágenes, tres de ellos correspondientes a Machine Learning y el cuarto a Deep Learning. Tras obtener los resultados de cada uno de ellos determinará que, para la base de datos usada, el método de Máquinas de Vectores de Soporte fue el que ofreció mejores resultados. Con un objetivo más elaborado que el de la anterior cita, Camargo et al. (2022) busca estudiar alternativas para automatizar el proceso de enseñanza y comunicación de los usuarios de la lengua de signos americana, usando algoritmos basados en redes neuronales y aprendizaje automático con el objetivo de generar un modelo inteligente que reconozca y clasifique las 27 señas que simbolizan las letras del abecedario en Lengua Americana de Señas (ASL). Para ello entrena y valida tres modelos ya utilizados en otros problemas de clasificación de imágenes basados en Redes Neuronales Convolucionales (CNN), en los cuales se exploran sistemáticamente ajustes en su estructura e híper-parámetros para identificar cuál es el modelo que mejor se adapta.

En su artículo, Dardas and Georganas (2011) presenta un sistema en tiempo real para la interacción con una aplicación mediante gestos de la mano. Se basa en detección y seguimiento de la mano desnuda en un fondo desordenado utilizando detección de piel y un algoritmo de comparación de contornos de mano, reconociendo gestos a través de una bolsa de características y una máquina de vectores de soporte multiclase (SVM). El entrenamiento está basado en extraer los puntos clave de cada imagen de entrenamiento utilizando el algoritmo SIFT (Scale-Invariant Feature Transform), una técnica de cuantificación de vectores que mapeará los puntos clave de cada imagen de entrenamiento en un vector de histogramas unificado dimensional (bag-of-words) tras K-means clustering. Este vector de histogramas se trata como vector de entrada de una SVM multiclase para construir el clasificador de entrenamiento. Con el mismo objetivo que el anterior, el trabajo de fin de máster de Palacios Fragoso (2021) se diferencia de él en que utiliza técnicas de machine learning y visión artificial. Al principio hace un estudio del software apropiado para las librerías de tratamiento de imágenes y las técnicas de machine learning, para luego implementar el reconocimiento de gestos. Al final concluye que de entre los modelos usados en el trabajo, las redes convolucionales obtienen los mejores resultados.

Nogales and Benalcázar (2021) propone una revisión bibliográfica sistemática del reconocimiento de gestos de la mano basado en información infrarroja y algoritmos de aprendizaje automático. Esta revisión es una versión extendida del trabajo presentado en la conferencia ICSE 2019. Para desarrollarla se ha utilizado la metodología Kitchenham. Se recupera información sobre las arquitecturas de los modelos, las técnicas implementadas en cada módulo, el tipo de aprendizaje utilizado (supervisado, no supervisado, semisupervisado y aprendizaje de refuerzo), y la clasificación de la precisión de reconocimiento, así como el tiempo de procesamiento.

El trabajo de Moreno Martín et al. (2016) sobre reconstrucción de imágenes podría ser usado en conjunto con nuestro sistema, permitiéndole así funcionar a partir de imágenes incompletas como entrada, que serían reconstruidas, y luego analizadas y clasificadas. Este trabajo se centra en descubrir y aprender patrones diferentes usando procesos gaussianos para la extrapolación de imágenes incompletas, permitiendo así su reconstrucción. Para ello usan "Kernels Superexpresivos", que son combinaciones de kernels más simples y que permiten al modelo aprender patrones muy complejos y dispares. Para contrarrestar el problema de las altas exigencias computacionales, se utilizó un conjunto de datos estructurado. El sistema fue implementado en Julia.

4 Desarrollo

Debido a que cada instancia de nuestra base de datos es una imagen, para obtener las salidas deseadas hemos clasificado las imágenes en 6 clases según el número de dedos levantados que muestran (0 a 5), usando para ello el nombre de archivo, como hemos indicado en la sección 2: Descripción del problema. Cada una de estas clases cuenta con 3.000 instancias: 1.500 de manos derechas y 1.500 de manos izquierdas.

4.1 Primera Aproximación

4.1.1 Descripción

Para llevar a cabo la primera aproximación, el dataset es el mismo que el mencionado en la sección 4: Desarrollo, en el que se usarán 3000 patrones para cada una de las 6 clases. De ellos, 1500 corresponden a la mano izquierda y 1500 a la derecha. Pero para esta aproximación **sólo utilizaremos dos clases: cero dedos levantados y un dedo levantado**, lo que daría un total de **6000 imágenes usadas**. Primero, delimitamos en cada instancia la zona en la que se encuentra la mano. Después, convertimos cada píxel de escala de grises a valores booleanos aplicando un umbral de 0.5 en un rango de 0 a 1 que, dependiendo de cuan claro u oscuro sea un píxel determinado, lo asocie a blanco (1, true) o negro (0, false) (ver figura 4).

Luego creamos un bounding box (función existente de Julia) rectangular que contenga todos los píxeles que estén a "true". Pero para evitar que se creen bounding boxes muy pequeños (debido a sombras, por ejemplo) se hace un proceso de eliminación de objetos de un tamaño menor de 100 píxeles cuadrados.

A partir de esto extraemos como característica la **proporción entre la anchura y altura del bounding box**, que permite diferenciar entre 0 dedos levantados, y 1 o más. No podemos usar esta característica con otras clases puesto que la diferencia en la proporción puede no ser lo suficientemente significativa para distinguir entre 3 y 4 dedos levantados, por ejemplo.

La figura 4 muestra una imagen de la clase 0 dedos levantados, y la tabla 1 muestra cómo se comporta la característica en estas clases.

No normalizaremos los datos ya que son proporciones, no tienen magnitud. Por otra parte, cabe destacar que se hará uso de la **validación cruzada de k iteraciones** para no depender de lo bien o mal que se hayan escogido los datos de entrenamiento y test.

Figure 4: Imagen binaria de 0 dedos

Clase	Media	Desviación
0 dedos	1.23	0.09
1 dedo	2.20	0.54

Table 1: Media y desv. típica de la proporción entre alto y ancho de los bounding boxes

4.1.2 Resultados

Con esta primera aproximación logramos diferenciar con éxito las manos que tienen 1 dedo levantado de las que no tienen ninguno (puños). Recordamos que para esta aproximación sólo hemos trabajado con las clases 0 dedos y 1 dedo levantados. La capas de entrada y de salida son comunes a todas las topologías probadas.

Para obtener los resultados de los modelos RRNNAA (tabla 2a) se usaron los parámetros que aparecen en la tabla, además de: la tasa de aprendizaje con valor 0.01, la función de transferencia sigmoidal, la proporción de validación 0.25, el número de entrenamientos que se realizarán dentro de cada iteración 20, el número máximo de ciclos que se entrenará cada RNA 500, y el número máximo de ciclos sin mejorar la pérdida de validación para realizar una parada temprana 20.

Para los resultados de los modelos SVM (tabla 2b) además de los parámetros kernel y c se usaron degree, con valor 3, y gamma, con valor 2.

Por otra parte, para los modelos de árboles de decisión y k-Nearest Neighbours, solo fueron usados los parámetros que aparecen en sus respectiva tablas (2c y 2d).

Modelo RRNNAA			
Nº	Topología	Precisión	
		Media	Desviación
1	(3)	95.78%	0.18%
2	(3,5)	93.62%	2.13%
3	(1,2)	83.64%	2.69%
4	(4)	95.48%	1.01%
5	(5,3)	94.58%	3.12%
6	(2,2)	95.27%	1.31%
7	(6)	96.08%	0.16%
8	(4,1)	94.89%	1.18%

(a) Resultados RRNNAA

Modelo SVM				
Nº	Kernel	C	Precisión	
			Media	Desviación
1	rbf	1	96.18%	0.14%
2	rbf	0.5	96.08%	0.18%
3	rbf	2	96.25%	0.21%
4	linear	0.5	96.08%	0.18%
5	linear	1	96.08%	0.18%
6	linear	1.5	96.08%	0.18%
7	poly	1	96.22%	0.19%
8	poly	1.5	96.22%	0.19%

(b) Resultados SVM

Modelo árboles de decisión			
Nº	Profundidad	Precisión	
		Media	Desviación
1	4	96.88%	0.40%
2	3	96.85%	0.41%
3	5	97.03%	0.54%
4	10	97.18%	0.53%
5	8	96.97%	0.52%
6	15	97.20%	0.48%

(c) Resultados árboles de decisión

Modelo k-Nearest Neighbors			
Nº	Valores de K	Precisión	
		Media	Desviación
1	3	97.07%	0.43%
2	2	97.00%	0.42%
3	5	97.00%	0.43%
4	8	96.90%	0.56%
5	10	96.78%	0.68%
6	7	96.82%	0.56%

(d) Resultados kNN

Table 2: Resultados primera aproximación

4.1.3 Discusión

Tras realizar las anteriores pruebas mostradas en la tabla 2 con los distintos modelos, pudimos comprobar que el **modelo que proporcionó mejores resultados** en esta primera aproximación para la métrica que buscamos mejorar (la precisión) fue el **árbol de decisión número 6**. Para seleccionar el mejor modelo nos hemos enfocado en obtener la **mayor media y menor desviación** posibles. Dado que las desviaciones no eran valores muy grandes, decidimos darle mayor peso a tener una mejor media. Además, seleccionaremos las configuraciones que mejores resultados dieron para cada uno de los modelos y con ellas realizaremos un nuevo entrenamiento con todo el conjunto de datos y, tras ello, realizaremos unas **pruebas con el conjunto de test** ya otorgado por la base de datos elegida, la cual consiste de **600 patrones por clase** (300 para la mano izquierda y otros 300 para la derecha), siendo para esta aproximación **1.200 imágenes en total**. Tras realizar las pruebas se obtuvieron las matrices de confusión representadas en la tabla 3.

	Negativos	Positivos
Negativos	561	39
Positivos	6	594

(a) Matriz de confusión del árbol de decisión número 6

	Negativos	Positivos
Negativos	561	39
Positivos	0	600

(b) Matriz de confusión del modelo kNN número 2

	Negativos	Positivos
Negativos	543	57
Positivos	0	600

(c) Matriz de confusión del modelo SVM número 3

	Negativos	Positivos
Negativos	541	59
Positivos	0	600

(d) Matriz de confusión del modelo RRNNAA número 7

Table 3: Matrices de confusión de los modelos con mejores resultados de la aproximación 1

Además conseguiremos unos nuevos valores de precisión para el conjunto de test usado, que se podrán corroborar con las anteriores tablas de confusión (tabla 3). Para el **modelo** que **mejor resultado dio** con la validación cruzada, **árbol de decisión-6**, se obtuvo una **precisión de un 96.25 %**. Por otra parte para el resto se consiguieron los siguientes valores de precisión: 96.91% para kNN-2, 96.75% para SVM-3, y 95.08% para ANN-7.

En este caso podemos observar cómo el modelo kNN-2 obtuvo una mejor precisión con el conjunto de test que el modelo de árboles de decisión 6, lo cual significa que para este conjunto de imágenes se comporta de una manera más acertada. Pero esto no tiene por qué traducirse en mejores resultados de forma general con otros conjuntos de imágenes.

Aparte de las pruebas anteriores, también hemos realizado una gráfica (figura 5) para **representar las precisiones medias** mostradas en la tabla 2.

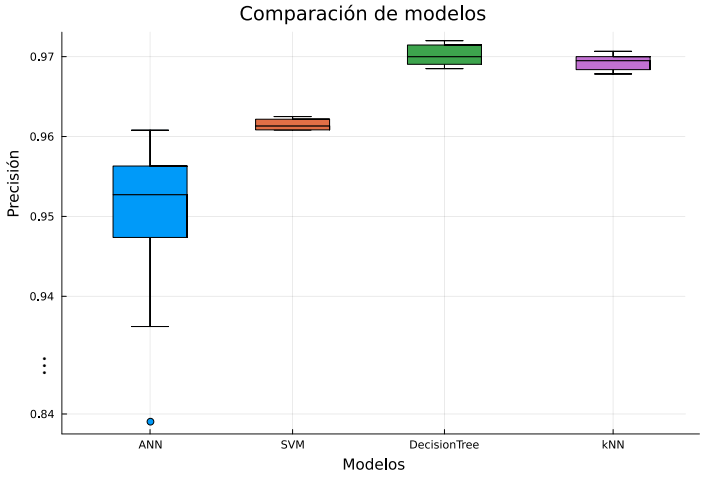

Figure 5: Diagrama de cajas de las precisiones obtenidas en la aproximación 1

En este diagrama de cajas (figura 5) podemos apreciar que el **modelo ANN** ha tenido los **resultados más dispares entre sí**, **dependiendo de** la **topología usada**. Además, algunos de los resultados llegan a desviarse bastante respecto al resto. En cuanto al resto de modelos, las precisiones obtenidas para cada uno de ellos son más homogéneas entre ellas. Finalmente, aunque se consiguieron buenas precisiones con todos los modelos, viendo la gráfica se pueden ordenar los distintos modelos por mejores resultados obtenidos. En orden de mejor a peor, serían: Árboles de decisión, kNN, SVM, y ANN.

4.2 Segunda Aproximación

4.2.1 Descripción

Para esta segunda aproximación añadiremos las clases de dos y tres dedos levantados y emplearemos una nueva característica que consistirá en obtener la **proporción entre píxeles a true y a false** dentro del bounding box (ver Figura 8). Esta característica nos permitirá diferenciar entre las manos que tengan 1, 2, y 3 dedos levantados. Además, junto con la anterior característica, nos permitirá en posteriores aproximaciones diferenciar 4 y 5 dedos levantados, ya que la anchura de la bounding box aumentará, y aunque las proporciones de la misma pudiesen llegar a parecerse a las de 0 dedos levantados, la nueva característica permitiría distinguir entre estas clases (teniendo 5 dedos más píxeles a false que la de 0 dedos).

El dataset es el mismo que el mencionado en la sección 4: Desarrollo, en la que se usarán **3.000 patrones por cada clase**, **1.500** para la **mano izquierda** y otros **1.500 para la derecha**. Para esta aproximación usaremos **en total cuatro clases**: cero, uno, dos y tres dedos levantados; lo que daría un **total de 12.000 imágenes usadas**.

La **normalización** que emplearemos es la de **mínimo/máximo**, puesto que buscamos que los **datos estén en un rango específico** (entre 0 y 1). Esto es esencial cuando se están normalizando características que tienen límites específicos (las proporciones del bounding box en nuestro caso), evitando así que se les dé más prioridad a unas que a otras. Elegimos esta normalización y no la de media 0 porque la de media 0 es útil cuando interesa la distribución relativa de los datos y se busca que tengan una media de 0 y una desviación típica de 1, lo cual NO es nuestro caso. Sería especialmente beneficioso en algoritmos que asumen que los datos siguen una distribución normal.

La validación cruzada empleada es la misma que en la anterior aproximación (k iteraciones).

Clase	Media	Desviación
0	1.23	0.09
1	2.20	0.54
2	2.02	0.34
3	1.71	0.22

(a) Media y desv. típica de la proporción entre alto y ancho de los bounding boxes

Clase	Media	Desviación
0	1.41	0.25
1	0.99	0.44
2	0.98	0.25
3	0.85	0.18

(b) Media y desv. típica de la proporción entre 1s y 0s de los bounding boxes

Table 4: Comportamiento de las características en función de la clase

(a) Imagen binaria de 2 dedos con bounding box

(b) Imagen binaria de 3 dedos con bounding box

Figure 6: Imágenes binarias con bounding box

4.2.2 Resultados

Los resultados obtenidos en esta aproximación vienen reflejados en la tabla 5. Los parámetros usados que no aparecen en la tabla serán los mismos que fueron descritos en la sección 4.1.2 (Resultados de la aproximación 1).

Para los modelos RRNNAA se usaron los parámetros que aparecen en la tabla (5a), además de: la tasa de aprendizaje con valor 0.01, la función de transferencia sigmoidal, la proporción de validación es 0.25, el número de entrenamientos que se realizan dentro de cada iteración es 20, el número máximo de ciclos que se entrena cada RRNNAA es 500, y el número máximo de ciclos sin mejorar la pérdida de validación para realizar una parada temprana es 20.

Para los resultados de los modelos SVM (tabla 5b) además de los parámetros kernel y c se usaron degree, con valor 3, y gamma, con valor 2.

Y para los modelos de árboles de decisión y k-Nearest Neighbours, solo fueron usados los parámetros que aparecen en sus respectiva tablas (5c y 5d).

Modelo RRNNAA			
Nº	Topología	Precisión	
		Media	Desviación
1	(3)	69.04%	0.57%
2	(3, 5)	59.88%	1.3%
3	(1, 2)	44.08%	3.04%
4	(4)	69.22%	1.79%
5	(5, 3)	65.41%	6.80%
6	(2, 2)	58.77%	4.80%
7	(6)	70.05%	0.63%
8	(4, 1)	52.06%	2.06%

(a) Resultados RRNNAA

Modelo SVM				
Nº	Kernel	C	Precisión	
			Media	Desviación
1	rbf	1	79.72%	0.83%
2	rbf	0.5	78.95%	0.92%
3	rbf	2	79.97%	0.75%
4	linear	0.5	67.83%	1.17%
5	linear	1	70.85%	0.95%
6	linear	1.5	71.07%	0.92%
7	poly	1	67.52%	0.95%
8	poly	1.5	72.80%	0.99%

(b) Resultados SVM

Modelo árboles de decisión			
Nº	Profundidad	Precisión	
		Media	Desviación
1	4	77.61%	0.87%
2	3	71.18%	0.51%
3	5	79.23%	0.57%
4	10	85.37%	0.91%
5	8	83.30%	0.50%
6	15	90.91%	0.67%

(c) Resultados árboles de decisión

Modelo k-Nearest Neighbors			
Nº	Valores de K	Precisión	
		Media	Desviación
1	3	82.84%	0.84%
2	2	86.15%	0.62%
3	5	84.64%	1.05%
4	8	83.23%	0.79%
5	10	83.50%	0.91%
6	7	83.42%	1.18%

(d) Resultados kNN

Table 5: Resultados segunda aproximación

4.2.3 Discusión

Tras realizar las anteriores pruebas con los modelos mostradas en la tabla 5, pudimos comprobar que el **modelo que proporcionó mejores resultados** en esta segunda aproximación para la métrica que buscamos mejorar (la precisión) fue de nuevo el **árbol de decisión número 6**. Para seleccionar el mejor modelo nos hemos fijado en que se obtuviese la **mayor media y la menor desviación posibles**. Además, seleccionaremos las configuraciones que mejores resultados dieron para cada uno de los modelos y con ellas realizaremos un nuevo entrenamiento con todo el conjunto de datos y, tras ello, unas pruebas con el conjunto de test ya otorgado por la base de datos elegida (la cual consiste de **600 patrones por clase**, 300 para la mano izquierda y otros 300 para la derecha, siendo para esta aproximación 2400 imágenes en total). Las pruebas dieron como resultado las matrices de confusión representadas en la tabla 6:

		Predicción			
		0	1	2	3
Real	0	484	94	22	0
	1	33	433	109	25
	2	6	106	371	117
	3	0	30	156	414

(a) Matriz de confusión del árbol de decisión número 6

		Predicción			
		0	1	2	3
Real	0	540	60	0	0
	1	33	466	90	11
	2	8	121	389	82
	3	0	33	210	357

(b) Matriz de confusión del modelo kNN número 2

		Predicción			
		0	1	2	3
Real	0	588	12	0	0
	1	45	440	107	8
	2	11	52	457	80
	3	0	10	169	421

(c) Matriz de confusión del modelo SVM número 3

		Predicción			
		0	1	2	3
Real	0	600	0	0	0
	1	90	460	20	30
	2	12	247	259	82
	3	0	19	231	350

(d) Matriz de confusión del modelo RRNNAA número 7

Table 6: Matrices de confusión de los modelos con mejores resultados de la aproximación 2

Como se puede observar en las matrices de confusión (tabla 6), las **imágenes con cero dedos levantados** fueron las **mejores clasificadas**. Esto puede haberse debido a que eran las **más distintas al resto** y, por ende, las más fáciles de diferenciar. Por otra parte, las **peores clasificadas**, de forma general, fueron las que tenían **dos dedos levantados**, quizás por **tener gran similitud con las imágenes de uno y tres dedos levantados**.

A continuación, hemos calculado **nuevos valores de precisión para el conjunto de test** usado, que se podrán corroborar con las anteriores tablas de confusión (tabla 6). Para el modelo que mejor resultado dio con la validación cruzada, el **árbol de decisión-6**, se **obtuvo una precisión de un 70.91 %**. Por otra parte, para el resto se consiguieron los siguientes valores de precisión: 73.00% para kNN-2, 79.41% para SVM-3, y 69.54% para ANN-7.

En este caso podemos observar cómo el **modelo SVM-3** obtuvo una mejor precisión con el conjunto de test que el modelo de árboles de decisión 6, lo cual significa que para este conjunto de imágenes se comporta de una manera más acertada. Pero esto no asegura que vaya a generalizar bien con otros conjuntos de imágenes.

Si comparamos estos resultados con los conseguidos en la aproximación uno (ver sección 4.1.3: Discusión), podemos comprobar como las **precisiones bajaron** considerablemente. Esto puede ser una **consecuencia de la adición de las dos nuevas clases**, lo que complica el problema a resolver.

Además de realizar las pruebas anteriores, también hemos creado un diagrama de cajas (figura 7) para las precisiones medias mostradas en la tabla 5:

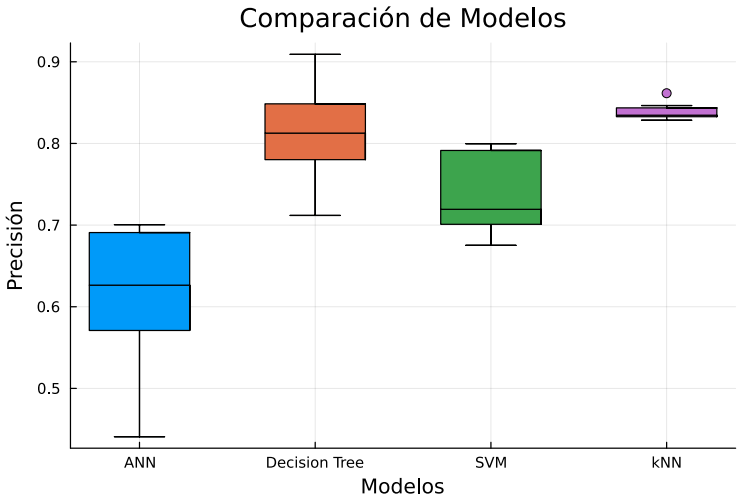

Figure 7: Diagrama de cajas de las precisiones obtenidas en la aproximación 2

En este diagrama podemos apreciar que el **modelo ANN** ha sido el que **peores resultados** obtuvo, llegando a tener un valor mínimo bastante bajo respecto al resto de modelos. Por otra parte, para los árboles de decisión y los SVM las precisiones obtenidas son algo más estables respecto a las del modelo ANN. Por último, los kNN son los que consiguieron unas precisiones más parecidas entre ellas. Por tanto, pese a que el modelo de árboles de decisión alcanzó un máximo mayor, es importante notar que los resultados que logran los kNN no dependen tanto de los parámetros usados.

4.3 Tercera Aproximación

4.3.1 Descripción

Para esta tercera aproximación realizamos una optimización en el momento de procesar las imágenes, la cual nos ayudará a obtener mejores resultados: hemos cambiado el **umbral de luminancia** (rango de 0 a 1) que era usado para transformar las imágenes a matrices binarias. El valor anterior, 0.5, pasa a ser 0.4, lo que hará que más valores oscuros sean ahora convertidos a 1/true, formando parte de la mano. Hemos comprobado que esto ayuda a identificar la mano de manera más precisa (ver figura 9).

También hemos notado que ciertos bounding boxes tomados para el entrenamiento no correspondían a la mano, sino a partes de ella. Para subsanarlo, se han **ordenado los bounding boxes** de cada instancia por **tamaño**, tomando el segundo de mayor tamaño como el correspondiente a la mano, ya que el primero pertenecería al fondo negro (ver figura 10).

A mayores, se ha **añadido una clase**: la de cuatro dedos levantados, teniendo ya un **total de cinco clases**: cero, uno, dos, tres y cuatro dedos levantados; lo cual resultaría en un total de **15000 imágenes usadas**. El dataset es el mismo que el mencionado en la sección 4: Desarrollo, en la que se usarán 3000 patrones por clase (1500 para la mano izquierda y otros 1500 para la derecha). En esta aproximación no se añadirá ninguna característica nueva ya que, como se mencionó en la sección 4.2.1: Descripción, las características implementadas hasta ahora logran diferenciar las clases mencionadas a un nivel suficiente.

La normalización empleada es la misma que en las demás aproximaciones (mínimo-máximo, entre 0 y 1) por las mismas razones. La validación cruzada también es la misma (k iteraciones).

Clase	Media	Desviación
0	1.24	0.08
1	2.12	0.17
2	2.01	0.19
3	1.67	0.18
4	1.26	0.09

(a) Media y desv. típica de la proporción entre alto y ancho de los bounding boxes

Clase	Media	Desviación
0	1.81	0.23
1	1.02	0.21
2	1.21	0.27
3	1.04	0.22
4	0.75	0.06

(b) Media y desv. típica de la proporción entre 1s y 0s de los bounding boxes

Table 7: Comportamiento de las características en función de la clase

(a) Imagen binaria de 2 dedos con bounding box

(b) Imagen binaria de 3 dedos con bounding box

Figure 8: Imágenes binarias con bounding box

(a) Imagen binaria de 2 dedos con bounding box de la aproximación 2

(b) Imagen binaria de 3 dedos con bounding box de la aproximación 2

(c) Imagen binaria de 2 dedos con bounding box de la aproximación 3

(d) Imagen binaria de 3 dedos con bounding box de la aproximación 3

Figure 9: Comparativa imágenes binarias con bounding box de la aproximación 2 y 3

(a) Imagen binaria de 3 dedos con bounding box de la aproximación 2

(b) Imagen binaria de 3 dedos con bounding box de la aproximación 3

Figure 10: Comparativa caso problemático imágenes binarias con bounding box

En el **caso problemático** de la **figura 10** ocurren los dos problemas que señalábamos anteriormente, el **dedo anular esta tomándose como un objeto aparte**, y el **bounding box que se usa para el procesado es el del dedo**, no el de la mano. **Gracias a los cambios de esta aproximación**, aún si el dedo no fuera tomado como parte de la mano o si el programa hubiera creado más bounding boxes (por ejemplo debido a sombras), **ahora se tomaría el segundo bounding box de mayor tamaño** para intentar que este corresponda a la mano (ignorando todos los bounding boxes de menor tamaño).

4.3.2 Resultados

Los resultados obtenidos en esta aproximación están reflejados en la tabla 8. Además, los parámetros usados que no aparecen en la tabla serán los mismos que fueron descritos en la sección 4.1.2: Resultados.

Modelo RRNNAA			
Nº	Topología	Precisión	
		Media	Desviación
1	(3)	79.51%	0.51%
2	(3, 5)	84.09%	0.50%
3	(1, 2)	63.58%	3.77%
4	(4)	80.56%	0.60%
5	(5, 3)	82.41%	1.51%
6	(2, 2)	75.11%	3.54%
7	(6)	81.96%	0.41%
8	(4, 1)	59.04%	4.09%

(a) Resultados RRNNAA

Modelo SVM				
Nº	Kernel	C	Precisión	
			Media	Desviación
1	rbf	1	85.90%	0.37%
2	rbf	0.5	85.95%	0.59%
3	rbf	2	86.04%	0.38%
4	linear	0.5	85.14%	0.59%
5	linear	1	85.30%	0.78%
6	linear	1.5	85.43%	0.68%
7	poly	1	84.97%	0.59%
8	poly	1.5	85.26%	0.66%

(b) Resultados SVM

Modelo árboles de decisión			
Nº	Profundidad	Precisión	
		Media	Desviación
1	4	81.79%	0.59%
2	3	80.37%	0.69%
3	5	85.73%	0.64%
4	10	92.53%	0.43%
5	8	90.19%	0.32%
6	15	97.60%	0.28%

(c) Resultados árboles de decisión

Modelo k-Nearest Neighbors			
Nº	Valores de K	Precisión	
		Media	Desviación
1	3	91.81%	0.15%
2	2	94.64%	0.55%
3	5	92.87%	0.32%
4	8	91.05%	0.31%
5	10	90.90%	0.27%
6	7	91.29%	0.29%

(d) Resultados kNN

Table 8: Resultados tercera aproximación

4.3.3 Discusión

Tras realizar las anteriores pruebas con los modelos mostrados en la tabla 8, pudimos comprobar que el **modelo que proporcionó mejores resultados** en esta aproximación para la métrica que buscamos mejorar (la precisión) fue el **árbol de decisión número 6**, por un margen significativo. Para realizar esta selección nos hemos fijado en que se obtuviese la mayor media y la menor desviación posibles. Además, seleccionaremos las configuraciones que mejores resultados dieron para cada uno de los modelos y con ellas realizaremos un nuevo entrenamiento con todo el conjunto de datos y, tras ello, unas pruebas con el conjunto de test ya otorgado por la base de datos elegida, la cual consiste de 600 patrones por clase (300 para la mano izquierda y 300 para la derecha), resultando para esta aproximación en 3000 imágenes en total. Tras las pruebas se obtuvieron la matrices de confusión representadas en la la tabla 9.

		Predicción				
		0	1	2	3	4
Real	0	600	0	0	0	0
	1	0	434	148	8	10
	2	0	160	330	110	0
	3	0	8	80	488	24
	4	0	8	0	8	584

(a) Matriz de confusión del árbol de decisión número 6

		Predicción				
		0	1	2	3	4
Real	0	600	0	0	0	0
	1	0	432	154	4	10
	2	0	134	366	100	0
	3	0	12	70	480	38
	4	0	6	0	16	578

(b) Matriz de confusión del modelo kNN número 2

		Predicción				
		0	1	2	3	4
Real	0	600	0	0	0	0
	1	0	416	168	6	10
	2	0	34	458	108	0
	3	0	2	30	544	24
	4	0	0	0	14	586

(c) Matriz de confusión del modelo SVM número 3

		Predicción				
		0	1	2	3	4
Real	0	600	0	0	0	0
	1	0	452	128	10	10
	2	0	84	448	62	6
	3	0	0	82	486	32
	4	0	0	0	20	580

(d) Matriz de confusión del modelo RRNNAA número 7

Table 9: Matrices de confusión de los modelos con mejores resultados de la aproximación 3

Viendo las matrices de confusión (tabla 9), se puede notar que las imágenes con cero y cuatro dedos levantados fueron muy bien clasificadas. Esto puede haberse debido a que eran las más distintas al resto. Por otra parte, las peores clasificadas, de forma general, volvieron a ser las que tenían dos dedos levantados. La razón podría ser la misma que la dicha anteriormente, que son las más confundibles por su alta similitud con las imágenes de uno y tres dedos levantados.

Los nuevos valores de precisión para el conjunto de test se podrán corroborar con las anteriores tablas de confusión (tabla 9). Para el **modelo que mejor resultado dio** con la validación cruzada, **árbol de decisión-6**, se **obtuvo una precisión de un 81.20%**. Por otra parte, para el resto se consiguieron los siguientes valores de precisión: 81.86% para kNN-2, 86.80% para SVM-3, y 85.53% para ANN-7.

En este caso podemos observar que el modelo SVM-3 obtuvo una mejor precisión con el conjunto de test que el modelo de árboles de decisión 6, lo cual significa que para este conjunto de imágenes se comporta de una manera más acertada. De nuevo recalcamos que esto no asegura que generalice bien con otros conjuntos de imágenes.

Si comparamos estos resultados con los conseguidos en la aproximación dos (ver sección 4.2.3: Discusión), podemos comprobar que las precisiones aumentaron. Esto puede ser debido a las mejoras aplicadas (umbral de luminancia, ordenación de bounding boxes).

Aparte de realizar las pruebas anteriores, también hemos creado una gráfica (figura 11) para las precisiones medias mostradas en la tabla 8.

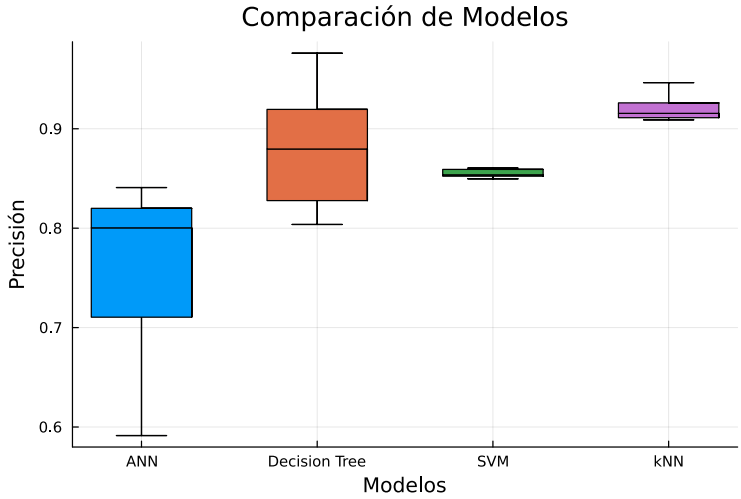

Figure 11: Diagrama de cajas de las precisiones obtenidas en la aproximación 3

Como podemos ver en este diagrama de cajas, los resultados para los modelos ANN, árboles de decisión y kNN están distribuidos de forma parecida a como se muestran en la gráfica de la anterior aproximación (figura 7), pero con unos valores generales mayores. Por otra parte, los **modelos SVM lograron precisiones menos dispares entre sí con respecto a la anterior aproximación**.

4.4 Cuarta Aproximación

4.4.1 Descripción

Para realizar la cuarta aproximación, el dataset es el mismo que el mencionado en la sección 4: Desarrollo. Se usarán 3000 patrones por clase: 1500 para la mano izquierda y otros 1500 para la derecha. Utilizamos en total seis clases: cero, uno, dos, tres, cuatro, y cinco dedos levantados; lo que daría un total de 18000 imágenes usadas (es decir, el dataset al completo).

Para esta aproximación añadiremos una nueva característica que consistirá en **trazar una línea horizontal en la imagen** a una altura relativa a la proporción del bounding box (que hemos establecido en 3/8 de la altura de la mano empezando por la punta de los dedos. Es decir, más cerca de la punta del dedo que de la base de la palma). Intentamos así que atraviese los dedos levantados para contar en la imagen binaria cuántas agrupaciones de 1s hay en esa línea de píxeles, lo que esencialmente nos dirá cuántas veces interseca con un dedo (o en general, con una parte de la mano). Usaremos esta característica para distinguir con mayor precisión entre las manos que tengan 1, 2, 3, y 4 dedos levantados, ya que son las que más clasificaciones erróneas producen.

La normalización empleada es la misma que en las demás aproximaciones (mínimo-máximo, entre 0 y 1) por las mismas razones. La validación cruzada también es la misma (k iteraciones).

Clase	Media	Desviación
0	1.81	0.23
1	1.02	0.21
2	1.21	0.18
3	1.68	0.46
4	1.84	0.25
5	1.10	0.18

(a) Media y desv. típica de la proporción entre alto y ancho de los bounding boxes

Clase	Media	Desviación
0	1.42	0.45
1	1.13	0.26
2	1.28	0.55
3	1.12	0.26
4	0.89	0.21
5	1.84	0.25

(b) Media y desv. típica de la proporción entre 1s y 0s de los bounding boxes

Clase	Media	Desviación
0	1.42	0.45
1	1.13	0.26
2	1.28	0.55
3	1.12	0.26
4	0.89	0.43
5	3.22	0.41

(c) Media y desv. típica de la proporción entre 1s y 0s de los bounding boxes

Table 10: Comportamiento de las características en función de la clase

4.4.2 Resultados

Los resultados obtenidos en esta aproximación están reflejados en la tabla 11. Los parámetros usados que no aparecen en la tabla serán los mismos que fueron descritos en la sección 4.1.2: Resultados.

Modelo RRNNAA			
Nº	Topología	Precisión	
		Media	Desviación
1	(3)	91.62%	0.52%
2	(3, 5)	91.79%	1.00%
3	(1, 2)	54.24%	3.48%
4	(4)	93.08%	0.26%
5	(5, 3)	87.32%	2.06%
6	(2, 2)	67.73%	2.13%
7	(6)	93.42%	0.56%
8	(4, 1)	48.31%	1.47%

(a) Resultados RRNNAA

Modelo SVM				
Nº	Kernel	C	Precisión	
			Media	Desviación
1	rbf	1	98.01 %	0.30%
2	rbf	0.5	98.11 %	0.26%
3	rbf	2	98.22 %	0.28%
4	linear	0.5	97.97 %	0.25%
5	linear	1	98.06 %	0.26%
6	linear	1.5	98.14 %	0.23%
7	poly	1	97.96 %	0.27%
8	poly	1.5	98.08 %	0.29%

(b) Resultados SVM

Modelo árboles de decisión			
Nº	Profundidad	Precisión	
		Media	Desviación
1	4	97.32 %	0.33%
2	3	79.57 %	0.52%
3	5	97.42 %	0.32%
4	10	99.34 %	0.09%
5	8	99.07 %	0.14%
6	15	99.66 %	0.11%

(c) Resultados árboles de decisión

Modelo k-Nearest Neighbors			
Nº	Valores de K	Precisión	
		Media	Desviación
1	3	98.86%	0.04%
2	2	99.20%	0.08%
3	5	99.10%	0.11%
4	8	98.79%	0.15%
5	10	98.79%	0.17%
6	7	98.83%	0.16%

(d) Resultados kNN

Table 11: Resultados cuarta aproximación

4.4.3 Discusión

Tras realizar las anteriores pruebas con los distintos modelos mostradas en la tabla 11, pudimos comprobar que el **modelo que proporcionó mejores resultados** en esta aproximación para la métrica que buscamos mejorar (la precisión) fue el **árbol de decisión número 6**, por un margen poco significativo. Para realizar esta selección nos hemos fijado en que se **obtuviese la mayor media y la menor desviación posibles**. Además, seleccionaremos las configuraciones que mejores resultados dieron para cada uno de los modelos y con ellas realizaremos un nuevo entrenamiento con todo el conjunto de datos y, tras ello, unas pruebas con el conjunto de test ya otorgado por la base de datos elegida, la cual consiste de **600 patrones por clase** (300 para la mano izquierda y 300 para la derecha), resultando para esta aproximación 3600 imágenes en total. Tras las pruebas se obtuvieron la matrices de confusión representadas en la la tabla 12.

		Predicción					
		0	1	2	3	4	5
Real	0	600	0	0	0	0	0
	1	0	588	12	0	0	0
	2	0	16	564	20	0	0
	3	0	4	30	558	8	0
	4	0	0	0	4	596	0
	5	0	0	0	0	0	600

(a) Matriz de confusión del árbol de decisión número 6

		Predicción					
		0	1	2	3	4	5
Real	0	600	0	0	0	0	0
	1	0	592	8	0	0	0
	2	0	32	552	16	0	0
	3	0	0	38	552	10	0
	4	0	0	0	0	600	0
	5	0	0	0	0	2	598

(b) Matriz de confusión del modelo kNN número 2

		Predicción					
		0	1	2	3	4	5
Real	0	600	0	0	0	0	0
	1	0	592	8	0	0	0
	2	0	2	578	20	0	0
	3	0	0	8	592	0	0
	4	0	0	0	2	598	0
	5	0	0	0	0	0	600

(c) Matriz de confusión del modelo SVM número 3

		Predicción					
		0	1	2	3	4	5
Real	0	592	0	0	0	0	0
	1	10	582	8	0	0	0
	2	0	2	578	20	0	0
	3	0	0	8	592	0	0
	4	0	0	0	4	548	48
	5	0	0	0	0	50	550

(d) Matriz de confusión del modelo RRNNAA número 7

Table 12: Matrices de confusión de los modelos con mejores resultados de la aproximación 4

Viendo las matrices de confusión (tabla 12), se puede notar que **todos los patrones fueron muy bien clasificados**. Lo cual difiere con el resto de aproximaciones, en las cuales siempre había clases que eran más fáciles de clasificar que otras. Esto es gracias a la **nueva característica añadida**, que fue descrita en la sección 4.3.1: Descripción.

Los nuevos valores de precisión para el conjunto de test usado se podrán corroborar con las anteriores tablas de confusión (tabla 12). Para el **modelo que mejor resultado dio** con la validación cruzada, **árbol de decisión-6**, se obtuvo una **precisión de un 97.38 %**. Para el resto de modelos se consiguieron los siguientes valores de precisión: 97.05% para kNN-2, 98.88% para SVM-3, y 95.61% para ANN-7.

En este caso podemos observar cómo el **modelo SVM-3 obtuvo una mejor precisión con el conjunto de test que el modelo de árboles de decisión 6**, lo cual significa que para este conjunto de imágenes se comporta de una manera más acertada. Pero esto no tiene por qué traducirse en mejores resultados de forma general con otros conjuntos de imágenes.

Si comparamos estos resultados con los conseguidos en la aproximación dos (ver sección 4.3.3: Discusión), podemos comprobar como las precisiones aumentaron. Esto puede ser debido a la nueva característica descrita en las sección 4.4.1: Descripción.

A parte de realizar las pruebas anteriores también hemos realizado un diagrama de cajas (Figura 12) para las precisiones medias mostradas en la tabla 11.

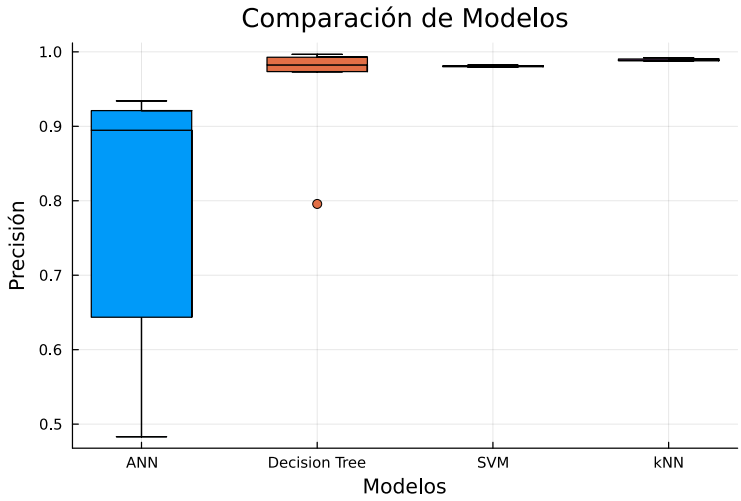

Figure 12: Diagrama de cajas de las precisiones obtenidas en la aproximación 4

En este diagrama de cajas (Figura 12) se puede observar cómo los **modelos ANN fueron los que peores resultados lograron** con respecto al resto, mientras que los **modelos de árboles de decisión, SVM y kNN consiguieron** obtener unas **precisiones altas y distribuidas de manera muy compacta** (menos para uno de los árboles de decisión, que obtuvo unos resultados que difieren con los del resto). Con esto podemos destacar que estos tres últimos modelos trabajan muy bien con las características y las imágenes usadas, consiguiendo clasificar de manera casi perfecta las imágenes dadas.

4.5 Aproximación Deep Learning

4.5.1 Descripción

Para realizar la aproximación de deep learning se ha utilizado una red de neuronas convolucional con el mismo dataset mencionado en la sección 4: Desarrollo. Para el entrenamiento se usarán **3000 patrones por clase**: **1500** para la **mano izquierda** y otros **1500 para la derecha**. Utilizamos en total **seis clases**: cero, uno, dos, tres, cuatro, y cinco dedos levantados; lo que daría un **total de 18000 imágenes usadas** (es decir, el dataset al completo). Por otra parte, para los test se usarán **600 patrones por clase**: **300** para la **mano izquierda** y otros **300 para la derecha**, que con las clases mencionadas anteriormente daría un **total de 3600 imágenes usadas**

El preprocesado esta vez ha sido igual a la última aproximación a excepción de la parte de extracción de características; es decir, las entradas son las imágenes pasadas a binario bajo el umbral de 0.4 comentado en la descripción 4.3.1. Este procesado es con el fin de que las instancias tengan el mínimo ruido/información innecesaria. Como preprocesado adicional, redujimos el tamaño de cada imagen de 128x128 píxeles a 50x50 píxeles, con el objetivo de facilitar el entrenamiento y reducir su duración. Inicialmente habíamos probado una reducción de tamaño mayor (28x28), pero la red era incapaz de mejorar su precisión y, tras varias pruebas, el valor óptimo que encontramos fue 50x50 píxeles.

Como contamos con un elevado número de patrones de entrenamiento (en este caso 18.000), en vez de entrenar pasando todos los patrones y modificando el error, dividiremos el conjunto de entrenamiento en subconjuntos (batches) y se irá aplicando uno a uno.

A pesar del tamaño de la base de datos, no hemos observado sobreentrenamiento alguno, por lo que no empleamos validación para entrenar las redes.

4.5.2 Resultados

Se utiliza en todas las arquitecturas el optimizador ADAM con una tasa de aprendizaje inicial de 0.001. Si no se mejora en 5 ciclos la precisión del el conjunto de entrenamiento, la tasa se divide entre 10. Si la tasa llega a valer menos de 1e-6 y no se mejora en 10 ciclos la precisión, el entrenamiento se para. La otra condición de parada del entrenamiento es si se alcanza una precisión de 99.9 % o mayor. Para todas las capas de convolución la función de transferencia utilizada es la *ReLU*.

CNN-1					
Tamaño batch			1024		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(4,4)	(2,2)	1=>32	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(4,4)	(2,2)	32=>16	N/A
4	MaxPool	N/A	N/A	N/A	(2,2)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			21		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.92 %			99.61 %		

Table 13: Arquitectura y resultados CNN-1

CNN-2					
Tamaño batch			1024		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(5,5)	(2,2)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>16	N/A
4	MaxPool	N/A	N/A	N/A	(2,2)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			48		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
98.54 %			98.66 %		

Table 14: Arquitectura y resultados CNN-2

En este caso, aún llegando a un máximo de precisión de 99.32 % en el ciclo 18, la CNN-2 (tabla 14) se estancó en 98.54 % en el ciclo 32 y se redujo varias veces la tasa de entrenamiento hasta 1e-7. Tras 10 ciclos sin mejoras se paró el entrenamiento.

CNN-3					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(5,5)	(2,2)	16=>16	N/A
4	MaxPool	N/A	N/A	N/A	(4,4)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			19		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.90 %			99.52 %		

Table 15: Arquitectura y resultados CNN-3

CNN-4					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(5,5)	(2,2)	1=>32	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	32=>16	N/A
4	MaxPool	N/A	N/A	N/A	(4,4)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			44		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
98.60 %			98.27 %		

Table 16: Arquitectura y resultados CNN-4

Se ha dado un caso similar al de CNN-2 (tabla 14): la CNN-4 (tabla 16), aún llegando a un máximo de precisión de 99.75 % en el ciclo 14, se estancó en 98.60 % en el ciclo 34 y se redujo varias veces la tasa de entrenamiento hasta 1e-7. Tras 10 ciclos sin mejorar, se paró el entrenamiento.

CNN-5					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>32	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	32=>16	N/A
4	MaxPool	N/A	N/A	N/A	(4,4)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			26		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.90 %			99.58 %		

Table 17: Arquitectura y resultados CNN-5

CNN-6					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>32	N/A
4	MaxPool	N/A	N/A	N/A	(4,4)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			27		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.91 %			99.52 %		

Table 18: Arquitectura y resultados CNN-6

CNN-7					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>16	N/A
4	MaxPool	N/A	N/A	N/A	(4,4)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			24		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.90 %			99.55 %		

Table 19: Arquitectura y resultados CNN-7

CNN-8					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>16	N/A
4	MaxPool	N/A	N/A	N/A	(2,2)
5	Flatten	N/A	N/A	N/A	N/A
6	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
7	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			36		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.90 %			99.50 %		

Table 20: Arquitectura y resultados CNN-8

CNN-9					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>16	N/A
4	MaxPool	N/A	N/A	N/A	(2,2)
5	Conv	(3,3)	(1,1)	16=>16	N/A
6	MaxPool	N/A	N/A	N/A	(2,2)
7	Flatten	N/A	N/A	N/A	N/A
8	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
9	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			18		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.91 %			99.55 %		

Table 21: Arquitectura y resultados CNN-9

CNN-10					
Tamaño batch			512		
Capas de la red					
Nº	Tipo	Filtro	Padding	Canales	Ventana
1	Conv	(3,3)	(1,1)	1=>16	N/A
2	MaxPool	N/A	N/A	N/A	(2,2)
3	Conv	(3,3)	(1,1)	16=>32	N/A
4	MaxPool	N/A	N/A	N/A	(2,2)
5	Conv	(3,3)	(1,1)	32=>32	N/A
6	MaxPool	N/A	N/A	N/A	(2,2)
7	Flatten	N/A	N/A	N/A	N/A
8	Totalmente conectada (clasificación)	N/A	N/A	N/A	N/A
9	softmax	N/A	N/A	N/A	N/A
Resultados					
Ciclos de entrenamiento			14		
Precisión					
Conjunto de entrenamiento			Conjunto de test		
99.95 %			99.61 %		

Table 22: Arquitectura y resultados CNN-10

4.5.3 Discusión

Tras realizar las anteriores pruebas con las distintas arquitecturas, pudimos comprobar que casi todas obtuvieron una precisión mayor al 99.00%, menos dos que obtuvieron una precisión mayor a 98.00% (ver tablas 14 y 16), la cual sigue siendo muy buena. De todas las arquitectura se puede destacar la usada en la tabla 22, ya que fue la que mejor precisión obtuvo en el menor número de ciclos realizados. En la tabla 23 se muestra la matriz de confusión para esta arquitectura. En ella podemos observar que se comporta de manera perfecta para las clases cero, uno, cuatro y cinco dedos levantados, mientras que difiere un poco para las clases dos y tres. Esto puede deberse a que las imágenes con dos y tres dedos levantados son las más parecidas.

		Predicción					
		0	1	2	3	4	5
Real	0	600	0	0	0	0	0
	1	0	600	0	0	0	0
	2	0	2	598	0	0	0
	3	0	0	12	588	0	0
	4	0	0	0	0	600	0
	5	0	0	0	0	0	600

Table 23: Matriz de confusión del CNN-10

5 Conclusiones

Reflexionando sobre el trabajo hecho hasta el momento, consideramos que los resultados son buenos y que el trabajo ha cumplido los objetivos iniciales. El hecho de que nuestras precisiones finales ronden el 95% refuerza esta afirmación.

Dados los resultados, creemos que este sistema es viable para resolver el problema de reconocimiento del número de dedos levantados en una imagen de una mano, y que podría extenderse a problemas derivados más complejos, como el reconocimiento de gestos. Otros sistemas como machine learning tendrían dificultades con este problema ya que los datos de entrada son imágenes, es decir, datos no estructurados; y el deep learning funciona mejor con este tipo de datos que con los estructurados.

Globalmente, el mejor modelo ha sido el árbol de decisión número 6, es decir, de 15 niveles de profundidad. En todas las aproximaciones ha tenido mejores resultados que los demás en cuanto a media (aprox. 99.67% en la última aproximación), y casi siempre por un margen significativo. En cuanto a desviación típica también ofrece resultados mejores que la mayoría de los otros modelos y arquitecturas, disminuyendo esta con cada nueva aproximación.

Los hiperparámetros usados para el árbol de decisión de 15 niveles de profundidad han sido: la tasa de aprendizaje con valor 0.01, la función de transferencia sigmoidal, la proporción de validación 0.25, el número de entrenamientos por iteración 20, el número máximo de ciclos a entrenar cada RNA 500, y el número máximo de ciclos sin mejorar la pérdida de validación para realizar una parada temprana 20.

Los resultados muestran que el modelo menos adecuado para este problema son las redes de neuronas artificiales, debido a la baja precisión media y alta desviación típica en la mayoría de configuraciones. Los demás modelos (SVM, árboles de decisión y k-nearest neighbours) producen resultados bastante similares, aunque en la última aproximación parece que el modelo k-nearest neighbours consiguió las menores desviaciones típicas, sólo superadas por los árboles de decisión de profundidades 8, 10 y 15, a falta de un test estadístico que lo confirme.

6 Trabajo futuro

Como ya se afirmó en la sección 5: Conclusiones, los resultados obtenidos fueron buenos y por lo tanto sería posible extenderse a problemas más complejos. Un ejemplo sería mejorar el sistema para que fuese posible detectar lenguaje de señas, lo cual permitiría que gente que no lo hable se pueda comunicar con personas mudas sin necesidad de que ellas tengan que escribir lo que desean decir. Una segunda línea de trabajo podría ser reconocimiento de gestos para control remoto de dispositivos; por ejemplo, interacción con una pantalla sin tocarla directamente, o realidad virtual sin necesidad de mandos (utilizando para ello una cámara que capture los movimientos de las manos del usuario).

Como un primer paso en esta dirección, podríamos complicar el problema quitando algunas de las restricciones declaradas en la sección 2: Descripción del problema, como por ejemplo que no todas las manos tengan la palma de cara a la cámara o que algunos dedos no estén levantados completamente.

References

Camargo, S. F. C., Canon, P. A. D. O. and Charry, P. O. J. P. (2022). Modelo de reconocimiento para la lengua de senas: Aproximacion comparativa entre métodos de reconocimiento de patrones por inteligencia artificial.
URL: *https://repository.urosario.edu.co/items/a0956efd-4d28-49e2-9fcf-3a8b8e105f60*

Chaudhary, A., Raheja, J. L., Das, K. and Raheja, S. (2013). Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey.
URL: *https://arxiv.org/abs/1303.2292*

Dardas, N. H. and Georganas, N. D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques *IEEE Transactions on Instrumentation and measurement* **60**(11), 3592–3607.
URL: *https://ieeexplore.ieee.org/document/5983442*

Feng, Z., Yang, B., Chen, Y., Zheng, Y., Xu, T., Li, Y., Xu, T. and Zhu, D. (2011). Features extraction from hand images based on new detection operators *Pattern Recognition* **44**(5), 1089–1105.
URL: *https://www.sciencedirect.com/science/article/pii/S0031320310003936*

Koryakin, P. (2018). Fingers. Accessed: 2024-02-29.
URL: *https://www.kaggle.com/datasets/koryakinp/fingers/*

Martín Rivas, F. (2018). *Detección de dedos utilizando técnicas de visión por computador y machine learning* B.S. thesis Universidad Carlos III de Madrid (UC3M).
URL: *https://e-archivo.uc3m.es/handle/10016/29590*

Moreno Martín, P. et al. (2016). *Extrapolación de imágenes mediante procesos glaussianos* B.S. thesis Universidad Autónoma de Madrid (UAM).
URL: *https://repositorio.uam.es/handle/10486/675443*

Nogales, R. and Benalcázar, M. (2021). Hand gesture recognition using machine learning and infrared information: a systematic literature review *International Journal of Machine Learning and Cybernetics* **12**.
URL: *https://link.springer.com/article/10.1007/s13042-021-01372-y*

Palacios Fragoso, M. (2021). Implementación de un sistema de reconocimiento de gestos mediante técnicas inteligentes.
URL: *https://ruc.udc.es/dspace/handle/2183/30392*